



Simba MySQL ODBC Driver

Installation and Configuration Guide

Simba Technologies Inc.

Version 1.0.9

November 15, 2018



Copyright © 2018 Simba Technologies Inc. All Rights Reserved.

Information in this document is subject to change without notice. Companies, names and data used in examples herein are fictitious unless otherwise noted. No part of this publication, or the software it describes, may be reproduced, transmitted, transcribed, stored in a retrieval system, decompiled, disassembled, reverse-engineered, or translated into any language in any form by any means for any purpose without the express written permission of Simba Technologies Inc.

Trademarks

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

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 MySQL ODBC Driver Installation and Configuration Guide* explains how to install and configure the Simba MySQL ODBC Driver. The guide also provides details related to features of the driver.

Audience

The guide is intended for end users of the Simba MySQL ODBC Driver, as well as administrators and developers integrating the driver.

Knowledge Prerequisites

To use the Simba MySQL ODBC Driver, the following knowledge is helpful:

- Familiarity with the platform on which you are using the Simba MySQL ODBC Driver
- Ability to use the data source to which the Simba MySQL ODBC Driver is connecting
- An understanding of the role of ODBC technologies and driver managers in connecting to a data source
- Experience creating and configuring ODBC 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 MySQL ODBC Driver	7
Windows Driver	8
Windows System Requirements	8
Installing the Driver on Windows	8
Creating a Data Source Name on Windows	9
Configuring Connection Options on Windows	11
Configuring Metadata Options on Windows	12
Configuring Cursor and Result Options on Windows	13
Configuring SSL Connections on Windows	14
Configuring Additional Driver Options on Windows	15
Configuring Logging Options on Windows	16
Verifying the Driver Version Number on Windows	18
macOS Driver	19
macOS System Requirements	19
Installing the Driver on macOS	19
Linux Driver	21
Linux System Requirements	21
Installing the Driver Using the Tarball Package	21
Verifying the Driver Version Number on Linux	22
Configuring the ODBC Driver Manager on Non-Windows Machines	23
Specifying ODBC Driver Managers on Non-Windows Machines	23
Specifying the Locations of the Driver Configuration Files	24
Configuring ODBC Connections on a Non-Windows Machine	26
Creating a Data Source Name on a Non-Windows Machine	26
Configuring a DSN-less Connection on a Non-Windows Machine	29
Configuring SSL Connections on a Non-Windows Machine	31
Configuring Logging Options on a Non-Windows Machine	33
Testing the Connection on a Non-Windows Machine	35
Using a Connection String	37
DSN Connection String Example	37
DSN-less Connection String Examples	37

Installation and Configuration Guide

Features	39
Data Types	39
Security and Authentication	42
Driver Configuration Options	43
Configuration Options Appearing in the User Interface	43
Configuration Options Having Only Key Names	60
Third-Party Trademarks	63
Third-Party Licenses	64

About the Simba MySQL ODBC Driver

The Simba MySQL ODBC Driver enables Business Intelligence (BI), analytics, and reporting on data that is stored in MySQL databases. The driver complies with the ODBC 3.80 data standard and adds important functionality such as Unicode, as well as 32- and 64-bit support for high-performance computing environments on all platforms.

ODBC 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 ODBC driver, which connects an application to the database. For more information about ODBC, see *Data Access Standards* on the Simba Technologies

website: <https://www.simba.com/resources/data-access-standards-glossary>. For complete information about the ODBC specification, see the *ODBC API Reference* from the Microsoft documentation: <https://docs.microsoft.com/en-us/sql/odbc/reference/syntax/odbc-api-reference>.

The *Installation and Configuration Guide* is suitable for users who are looking to access MySQL data from their desktop environment. Application developers might also find the information helpful. Refer to your application for details on connecting via ODBC.

 **Note:**

For information about how to use the driver in various BI tools, see the *Simba ODBC Drivers Quick Start Guide for Windows*: http://cdn.simba.com/docs/ODBC_QuickstartGuide/content/quick_start/intro.htm.

Windows Driver

Windows System Requirements

The Simba MySQL ODBC Driver supports MySQL versions 5.6 and 5.7.

Install the driver on client machines where the application is installed. Before installing the driver, make sure that you have the following:

- Administrator rights on your machine.
- A machine that meets the following system requirements:
 - One of the following operating systems:
 - Windows 10, 8.1, or 7 SP1
 - Windows Server 2016, 2012, or 2008 R2 SP1
 - 150 MB of available disk space

Before the driver can be used, the Visual C++ Redistributable for Visual Studio 2015 with the same bitness as the driver must also be installed. If you obtained the driver from the Simba website, then your installation of the driver automatically includes this dependency. Otherwise, you must install the redistributable manually. You can download the installation packages for the redistributable at <https://www.microsoft.com/en-ca/download/details.aspx?id=48145>.

Installing the Driver on Windows

On 64-bit Windows operating systems, you can execute both 32- and 64-bit applications. However, 64-bit applications must use 64-bit drivers, and 32-bit applications must use 32-bit drivers. Make sure that you use a driver whose bitness matches the bitness of the client application:

- `SimbaMySQLODBC32.msi` for 32-bit applications
- `SimbaMySQLODBC64.msi` for 64-bit applications

You can install both versions of the driver on the same machine.

To install the Simba MySQL ODBC Driver on Windows:

1. Depending on the bitness of your client application, double-click to run **SimbaMySQLODBC32.msi** or **SimbaMySQLODBC64.msi**.
2. Click **Next**.
3. Select the check box to accept the terms of the License Agreement if you agree, and then click **Next**.

4. To change the installation location, click **Change**, then browse to the desired folder, and then click **OK**. To accept the installation location, click **Next**.
5. Click **Install**.
6. When the installation completes, click **Finish**.

Creating a Data Source Name on Windows

Typically, after installing the Simba MySQL ODBC Driver, you need to create a Data Source Name (DSN).

Alternatively, for information about DSN-less connections, see [Using a Connection String](#) on page 37.

To create a Data Source Name on Windows:

1. Open the ODBC Data Source Administrator:
 - If you are using a 32-bit driver on a 64-bit machine, then run this application: `C:\Windows\SysWOW64\odbcad32.exe`.
 - Otherwise, open the **Control Panel**, then select **Administrative Tools**, and then double-click **Data Sources (ODBC)**.

 **Note:**

Make sure to select the ODBC Data Source Administrator that has the same bitness as the client application that you are using to connect to MySQL.

2. In the ODBC Data Source Administrator, click the **Drivers** tab, and then scroll down as needed to confirm that the Simba MySQL ODBC Driver appears in the alphabetical list of ODBC drivers that are installed on your system.
3. Choose one:
 - To create a DSN that only the user currently logged into Windows can use, click the **User DSN** tab.
 - Or, to create a DSN that all users who log into Windows can use, click the **System DSN** tab.

 **Note:**

It is recommended that you create a System DSN instead of a User DSN. Some applications load the data using a different user account, and might not be able to detect User DSNs that are created under another user account.

4. Click **Add**.

5. In the Create New Data Source dialog box, select **Simba MySQL ODBC Driver** and then click **Finish**. The Simba MySQL ODBC Driver DSN Setup dialog box opens.
6. In the **Data Source Name** field, type a name for your DSN.
7. Optionally, in the **Description** field, type relevant details about the DSN.
8. In the **User** field, type an appropriate user name for accessing the MySQL database.
9. If you are required to authenticate the connection using the native or SHA-256 authentication plugin, then in the **Password** field, type the password corresponding to the user name you typed above.
10. Optionally, if you are authenticating the connection using the SHA-256 authentication plugin, specify an RSA public key for encrypting your password by doing the following:
 - a. Select the **SSL** tab.
 - b. In the **RSA Public Key** field, specify the full path and name of the file containing the key.

 **Note:**

If you do not specify an RSA public key or configure SSL for your connection, the driver automatically retrieves a public key from the server and uses it to encrypt your password.

11. In the **Host** field, type the name or IP address of the MySQL server.
12. In the **Port** field, type the number of the TCP port that the server uses to listen for client connections.

 **Note:**

The default port used by MySQL is 3306.

13. In the **Catalog** field, type the name of the database that you want to connect to by default.

 **Note:**

You can still issue queries on other databases by specifying the database schema in your query statement.

14. To configure logging behavior for the driver, click **Logging Options**. For more information, see [Configuring Logging Options on Windows](#) on page 16.
15. To configure additional driver options, select a tab:
 - Select the **Connection** tab to configure how the driver handles connections and specify driver behavior that applies at connection time. For more

information, see [Configuring Connection Options on Windows](#) on page 11.

- Select the **Metadata** tab to configure how the driver handles specific types of data. For more information, see [Configuring Metadata Options on Windows](#) on page 12.
 - Select the **Cursor/Results** tab to configure how the driver handles cursors and query results. For more information, see [Configuring Cursor and Result Options on Windows](#) on page 13.
 - Select the **SSL** tab to configure encryption and identity verification using SSL. For more information, see [Configuring SSL Connections on Windows](#) on page 14.
 - Select the **Misc** tab to configure other additional driver options. For more information, see [Configuring Additional Driver Options on Windows](#) on page 15.
16. To test the connection, click **Test**. Review the results as needed, and then click **OK**.

 **Note:**

If the connection fails, then confirm that the settings in the Simba MySQL ODBC Driver DSN Setup dialog box are correct. Contact your MySQL server administrator as needed.

17. To save your settings and close the Simba MySQL ODBC Driver DSN Setup dialog box, click **OK**.
18. To close the ODBC Data Source Administrator, click **OK**.

Configuring Connection Options on Windows

You can configure options to modify how the driver handles connections and specify driver behavior that applies at connection time.

To configure connection options on Windows:

1. To access the connection options, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, and then select the **Connection** tab.
2. To compress the communication between the client and the server, select the **Use Compression** check box.

 **Note:**

Compression may reduce the amount of network traffic for certain workflows, but in some cases it may increase CPU usage.

3. To automatically reconnect to the server when a communication link error occurs, select the **Enable Automatic Reconnect** check box.
4. To prevent the driver from opening dialog boxes to prompt for additional information during connection time, select the **Don't Prompt When Connecting** check box.
5. To enable support for batched statements, select the **Allow Multiple Statements** check box.
6. To allow sandboxed connections through which you can reset the password after attempting to connect using an expired password, select the **Can Handle Expired Password** check box.
7. To close the connection after it has been inactive for the amount of time specified by the `interactive_timeout` setting on the server, select the **Interactive Client** check box.
8. In the **Initial Statement** field, type a statement for the driver to execute immediately after connecting to the server.
9. To save your settings and close the Simba MySQL ODBC Driver DSN Setup dialog box, click **OK**.

Configuring Metadata Options on Windows

You can configure how the driver handles specific types of data.

To configure metadata options on Windows:

1. To access the metadata options, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, and then select the **Metadata** tab.
2. To return BIGINT data as SQL_INTEGER data instead of SQL_BIGINT data, select the **Treat BIGINT Columns As INT Columns** check box.
3. To return Binary columns as Character columns, select the **Always Handle Binary Function Results As Character Data** check box.
4. To allow column names in queries to include the database name (for example, using the format `[Database].[Table].[Column]`), select the **Ignore Schema In Column Specifications** check box.

Note:

If this option is disabled, the driver returns an error if a column name specified in a statement includes the database name.

5. To return fully-qualified column names that include the table name when the `SQLDescribeCol()` function is called, select the **Include Table Name In SQLDescribeCol()** check box.

6. To limit column sizes to 32-bit signed values, select the **Limit column size to a signed 32-bit value** checkbox.
7. Optionally, to configure the driver to recognize table type information from the data source, select the **Enable Table Types** checkbox. For more information, see [Enable Table Types](#) on page 48.
8. To save your settings and close the Simba MySQL ODBC Driver DSN Setup dialog box, click **OK**.

Configuring Cursor and Result Options on Windows

You can configure how the driver handles cursors and query results.

To configure cursor and result options on Windows:

1. To access the cursor and result options, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, and then select the **Cursor/Results** tab.
2. To reduce memory consumption by preventing the driver from caching result sets into memory, select the **Don't Cache Results Of Forward-Only Cursors** check box.
3. To specify what the driver returns after DML operations, do one of the following:
 - To return the number of matched rows, select the **Return Matched Rows Instead Of Affected Rows** check box.
 - Or, to return the number of affected rows, clear the **Return Matched Rows Instead Of Affected Rows** check box.
4. To respect the `sql_auto_is_null` setting specified in the server instead of automatically disabling it, select the **Enable SQL_AUTO_IS_NULL** check box.

Note:

For more information about `sql_auto_is_null`, see "Server System Variables" in the *MySQL Reference Manual*:

https://dev.mysql.com/doc/refman/5.7/en/server-systemvariables.html#sysvar_sql_auto_is_null.

5. To specify how the driver handles date values where the month or day is 0, do one of the following:
 - To translate zero dates to a date value that is supported by ODBC, select the **Return Minimal Date For Zero Date** check box.
 - To return zero dates as NULL, clear the **Return Minimal Date For Zero Date** check box.

6. To save your settings and close the Simba MySQL ODBC Driver DSN Setup dialog box, click **OK**.

Configuring SSL Connections on Windows

If you are connecting to a MySQL server that has Secure Sockets Layer (SSL) enabled, then you can configure the driver to connect to an SSL-enabled socket and encrypt the connection. When connecting to a server over SSL, the driver supports identity verification between the client and the server.

Configuring an SSL Connection without Identity Verification

You can configure a connection that is encrypted by SSL but does not verify the identity of the client or the server.

To configure an SSL connection without verification on Windows:

1. To access the SSL options, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, and then select the **SSL** tab.
2. From the **SSL Mode** drop-down list, select one of the following options:
 - To use SSL encryption only if the server supports it, select **PREFERRED**.
 - Or, to require SSL encryption for the connection, select **REQUIRED**. If the server does not support SSL, the connection fails.
3. Optionally, in the **SSL Cipher** field, type a comma-separated list of permitted ciphers for encrypting the connection.
4. To specify the minimum version of SSL to use, from the **Minimum TLS** drop-down list, select the minimum version of SSL.
5. To have the driver use the Windows trust store, select the **Use Truststore** checkbox.
6. To save your settings and close the Simba MySQL ODBC Driver DSN Setup dialog box, click **OK**.

Configuring SSL Identity Verification

You can configure one-way verification so that the client verifies the identity of the MySQL server, or you can configure two-way verification so that the client and the sever both verify each other.

In both cases, you must provide a root certificate from a trusted certificate authority (CA) that the driver can use to check the server's certificate. If you are using two-way verification, then you must also provide a certificate that proves the identity of the client and a private key that encrypts the client certificate.

To configure SSL identity verification on Windows:

1. To access the SSL options, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, and then select the **SSL** tab.
2. To specify one or more root certificates from trusted CAs that you want to use to verify the server certificate, do one of the following:
 - To use a specific root certificate, in the **SSL Certificate Authority** field, specify the full path and name of the `.pem` file containing the certificate.
 - Or, to provide multiple root certificates, in the **SSL CA Path** field, specify the full path and name of the directory that contains the certificates. The driver uses the first valid certificate that it finds in the directory.
3. If two-way identity verification is necessary, do the following:
 - a. In the **SSL Key** field, specify the full path and name of the file that contains the private key used for encrypting the client certificate.
 - b. In the **SSL Certificate** field, specify the full path and name of the `.pem` file containing the certificate used for proving the identity of the client.
4. From the **SSL Mode** drop-down list, select one of the following options:
 - To use SSL encryption and identity verification only if the server supports it, select **VERIFY_CA**.
 - Or, to require SSL encryption and identity verification for the connection, select **VERIFY_IDENTITY**. If the server does not support SSL or if identity verification fails, the connection fails.
5. Optionally, in the **SSL Cipher** field, type a comma-separated list of permitted ciphers for encrypting the connection.
6. To specify the minimum version of SSL to use, from the **Minimum TLS** drop-down list, select the minimum version of SSL.
7. To have the driver use the Windows trust store, select the **Use Truststore** checkbox.
8. To save your settings and close the Simba MySQL ODBC Driver DSN Setup dialog box, click **OK**.

Configuring Additional Driver Options on Windows

You can configure driver options to modify the behavior of the driver.

To configure additional driver options on Windows:

1. To access the additional driver options, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, and then select the **Misc** tab.

2. To ignore spaces between function names and parentheses (() in statements so that function names are treated as keywords, select the **Ignore Space After Function Names** check box.
3. To disable support for transactions, select the **Disable Transaction Support** check box.
4. To translate the minimum ODBC date value to the MySQL zero date value when binding parameters, select the **Bind Minimal Date As Zero Date** check box.

 **Note:**

Enabling this option prevents statements from failing because of what seems to be a mismatch in the date values.

5. To emulate prepared statements on the client side, select the **Prepare Statements On The Client** check box.
6. To save your settings and close the Simba MySQL ODBC Driver DSN Setup dialog box, click **OK**.

Configuring Logging Options on Windows

To help troubleshoot issues, you can enable logging. In addition to functionality provided in the Simba MySQL ODBC Driver, the ODBC Data Source Administrator provides tracing functionality.

! Important:

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

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

To enable driver logging on Windows:

1. To access logging options, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, and then click **Logging Options**.
2. From the **Log Level** drop-down list, select the logging level corresponding to the amount of information that you want to include in log files:

Logging Level	Description
OFF	Disables all logging.

Logging Level	Description
FATAL	Logs severe error events that lead the driver to abort.
ERROR	Logs error events that might allow the driver to continue running.
WARNING	Logs events that might result in an error if action is not taken.
INFO	Logs general information that describes the progress of the driver.
DEBUG	Logs detailed information that is useful for debugging the driver.
TRACE	Logs all driver activity.

- In the **Log Path** field, specify the full path to the folder where you want to save log files.
- In the **Max Number Files** field, type the maximum number of log files to keep.

 **Note:**

After the maximum number of log files is reached, each time an additional file is created, the driver deletes the oldest log file.

- In the **Max File Size** field, type the maximum size of each log file in megabytes (MB).

 **Note:**

After the maximum file size is reached, the driver creates a new file and continues logging.

- Click **OK**.
- Restart your ODBC application to make sure that the new settings take effect.

The Simba MySQL ODBC Driver produces the following log files at the location you specify in the Log Path field:

- A `simbamysqlodbcdriver.log` file that logs driver activity that is not specific to a connection.

- A `simbamysqlodbcdriver_connection_[Number].log` file for each connection made to the database, where `[Number]` is a number that identifies each log file. This file logs driver activity that is specific to the connection.

If you enable the `UseLogPrefix` connection property, the driver prefixes the log file name with the user name associated with the connection and the process ID of the application through which the connection is made. For more information, see [UseLogPrefix](#) on page 61.

To disable driver logging on Windows:

1. Open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, and then click **Logging Options**.
2. From the **Log Level** drop-down list, select **LOG_OFF**.
3. Click **OK**.
4. Restart your ODBC application to make sure that the new settings take effect.

Verifying the Driver Version Number on Windows

If you need to verify the version of the Simba MySQL ODBC Driver that is installed on your Windows machine, you can find the version number in the ODBC Data Source Administrator.

To verify the driver version number on Windows:

1. Open the ODBC Administrator:
 - If you are using a 32-bit driver on a 64-bit machine, then run this application: `C:\Windows\SysWOW64\odbcad32.exe`.
 - Otherwise, open the **Control Panel**, then select **Administrative Tools**, and then double-click **Data Sources (ODBC)**.

Note:

Make sure to select the ODBC Data Source Administrator that has the same bitness as the client application that you are using to connect to MySQL.

2. Click the **Drivers** tab and then find the Simba MySQL ODBC Driver in the list of ODBC drivers that are installed on your system. The version number is displayed in the **Version** column.

macOS Driver

macOS System Requirements

The Simba MySQL ODBC Driver supports MySQL versions 5.6 and 5.7.

Install the driver on client machines where the application is installed. Each client machine that you install the driver on must meet the following minimum system requirements:

- macOS version 10.11, 10.12, or 10.13
- 250 MB of available disk space
- iODBC 3.52.9, 3.52.10, 3.52.11, or 3.52.12

Installing the Driver on macOS

The macOS version of the Simba MySQL ODBC Driver is delivered in a tarball named `SimbaMySQLODBC_[Version]-OSX.tar.gz`, where *[Version]* is the version number of the driver.

To install the Simba MySQL ODBC Driver on a macOS machine, you must do the following:

1. Create the installation directory by extracting the driver files from the tarball and copying them to the appropriate locations. For more information, see [Creating the Installation Directory on macOS](#) on page 19.
2. Configure the ODBC driver manager on your machine to work with the driver. For more information, see [Configuring the ODBC Driver Manager on Non-Windows Machines](#) on page 23.

Creating the Installation Directory on macOS

Create the installation directory for the Simba MySQL ODBC Driver on your macOS machine by extracting the driver files from the tarball and copying them to the appropriate locations.

To create the installation directory on macOS:

1. Create the following directory: `/Library/simba/mysql`

Note:

- If necessary, you can create an installation directory with different folder structure and naming. The paths given above are examples based on the default installation directories used by Simba drivers on macOS.
- In the steps below, this documentation uses the variable `[InstallDir]` to refer to this path.

2. Extract the `SimbaMySQLODBC-[Version]-OSX.tar.gz` file into a convenient temporary location.
3. In the `SimbaMySQLODBC-[Version]-OSX` folder, extract the `SimbaMySQLODBC-[Version].tar.gz` file.
4. From the `SimbaMySQLODBC-[Version]` folder, copy the `ErrorMessages` and `lib` subfolders into `[InstallDir]`.
5. From the `SimbaMySQLODBC-[Version]-OSX` subfolder that you extracted during step 2, copy the following files and folders to the `[InstallDir]/lib` directory:
 - The `MySQLODBC.did` file.
 - From the `setup` subfolder, the `odbc.ini`, `odbcinst.ini`, and `simba.mysqlodbc.ini` files.

You should now have the following folders in `[InstallDir]`:

- `ErrorMessages`
- `lib`

The `lib` subfolder should contain the following files and folders:

- `libmysqlodbc_sbu.dylib`
- `MySQLODBC.did`
- `odbc.ini`
- `odbcinst.ini`
- `simba.mysqlodbc.ini`

Next, configure the ODBC driver manager on your machine to work with the driver. For more information, see [Configuring the ODBC Driver Manager on Non-Windows Machines](#) on page 23.

Linux Driver

Linux System Requirements

The Simba MySQL ODBC Driver supports MySQL versions 5.6 and 5.7.

Install the driver on client machines where the application is installed. Each client machine that you install the driver on must meet the following minimum system requirements:

- One of the following distributions:
 - Red Hat® Enterprise Linux® (RHEL) 6 or 7
 - CentOS 6 or 7
 - SUSE Linux Enterprise Server (SLES) 11 or 12
 - Debian 8 or 9
 - Ubuntu 14.04, 16.04, or 18.04
- 90 MB of available disk space
- One of the following ODBC driver managers installed:
 - iODBC 3.52.9, 3.52.10, 3.52.11, or 3.52.12
 - unixODBC 2.3.2, 2.3.3, or 2.3.4

To install the driver, you must have root access on the machine.

Installing the Driver Using the Tarball Package

The Simba MySQL ODBC Driver is available as a tarball package named `SimbaMySQLODBC-[Version].[Release]-Linux.tar.gz`, where *[Version]* is the version number of the driver and *[Release]* is the release number for this version of the driver. The package contains both the 32-bit and 64-bit versions of the driver.

On 64-bit editions of Linux, you can execute both 32- and 64-bit applications. However, 64-bit applications must use 64-bit drivers, and 32-bit applications must use 32-bit drivers. Make sure that you use a driver whose bitness matches the bitness of the client application. You can install both versions of the driver on the same machine.

To install the driver using the tarball package:

1. Log in as the root user, and then navigate to the folder containing the tarball package.
2. Run the following command to extract the package and install the driver:

```
tar -zxvf [TarballName]
```

Where *[TarballName]* is the name of the tarball package containing the driver.

The Simba MySQL ODBC Driver files are installed in the `opt/simba/mysql` directory.

Next, configure the environment variables on your machine to make sure that the ODBC driver manager can work with the driver. For more information, see [Configuring the ODBC Driver Manager on Non-Windows Machines](#) on page 23.

Verifying the Driver Version Number on Linux

If you need to verify the version of the Simba MySQL ODBC Driver that is installed on your Linux machine, you can query the version number through the command-line interface if the driver was installed using an RPM file.

To verify the driver version number on Linux:

- Depending on your package manager, at the command prompt, run one of the following commands:

The command returns information about the Simba MySQL ODBC Driver that is installed on your machine, including the version number.

Configuring the ODBC Driver Manager on Non-Windows Machines

To make sure that the ODBC driver manager on your machine is configured to work with the Simba MySQL ODBC Driver, do the following:

- Set the library path environment variable to make sure that your machine uses the correct ODBC driver manager. For more information, see [Specifying ODBC Driver Managers on Non-Windows Machines](#) on page 23.
- If the driver configuration files are not stored in the default locations expected by the ODBC driver manager, then set environment variables to make sure that the driver manager locates and uses those files. For more information, see [Specifying the Locations of the Driver Configuration Files](#) on page 24.

After configuring the ODBC driver manager, you can configure a connection and access your data store through the driver. For more information, see [Configuring ODBC Connections on a Non-Windows Machine](#) on page 26.

Specifying ODBC Driver Managers on Non-Windows Machines

You need to make sure that your machine uses the correct ODBC driver manager to load the driver. To do this, set the library path environment variable.

macOS

If you are using a macOS machine, then set the `DYLD_LIBRARY_PATH` environment variable to include the paths to the ODBC driver manager libraries. For example, if the libraries are installed in `/usr/local/lib`, then run the following command to set `DYLD_LIBRARY_PATH` for the current user session:

```
export DYLD_LIBRARY_PATH=$DYLD_LIBRARY_PATH:/usr/local/lib
```

For information about setting an environment variable permanently, refer to the macOS shell documentation.

Linux

If you are using a Linux machine, then set the `LD_LIBRARY_PATH` environment variable to include the paths to the ODBC driver manager libraries. For example, if the libraries are installed in `/usr/local/lib`, then run the following command to set `LD_LIBRARY_PATH` for the current user session:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
```

For information about setting an environment variable permanently, refer to the Linux shell documentation.

Specifying the Locations of the Driver Configuration Files

By default, ODBC driver managers are configured to use hidden versions of the `odbc.ini` and `odbcinst.ini` configuration files (named `.odbc.ini` and `.odbcinst.ini`) located in the home directory, as well as the `simba.mysqlodbc.ini` file in the `lib` subfolder of the driver installation directory. If you store these configuration files elsewhere, then you must set the environment variables described below so that the driver manager can locate the files.

If you are using iODBC, do the following:

- Set `ODBCINI` to the full path and file name of the `odbc.ini` file.
- Set `ODBCINSTINI` to the full path and file name of the `odbcinst.ini` file.
- Set `SIMBA_MYSQL_ODBC_INI` to the full path and file name of the `simba.mysqlodbc.ini` file.

If you are using unixODBC, do the following:

- Set `ODBCINI` to the full path and file name of the `odbc.ini` file.
- Set `ODBCSYSINI` to the full path of the directory that contains the `odbcinst.ini` file.
- Set `SIMBA_MYSQL_ODBC_INI` to the full path and file name of the `simba.mysqlodbc.ini` file.

For example, if your `odbc.ini` and `odbcinst.ini` files are located in `/usr/local/odbc` and your `simba.mysqlodbc.ini` file is located in `/etc`, then set the environment variables as follows:

For iODBC:

```
export ODBCINI=/usr/local/odbc/odbc.ini
export ODBCINSTINI=/usr/local/odbc/odbcinst.ini
export SIMBA_MYSQL_ODBC_INI=/etc/simba.mysqlodbc.ini
```

For unixODBC:

```
export ODBCINI=/usr/local/odbc/odbc.ini
export ODBCSYSINI=/usr/local/odbc
```

```
export SIMBA_MYSQL_ODBC_INI=/etc/simba.mysqlodbc.ini
```

To locate the `simba.mysqlodbc.ini` file, the driver uses the following search order:

1. If the `SIMBA_MYSQL_ODBC_INI` environment variable is defined, then the driver searches for the file specified by the environment variable.
2. The driver searches the directory that contains the driver library files for a file named `simba.mysqlodbc.ini`.
3. The driver searches the current working directory of the application for a file named `simba.mysqlodbc.ini`.
4. The driver searches the home directory for a hidden file named `.simba.mysqlodbc.ini` (prefixed with a period).
5. The driver searches the `/etc` directory for a file named `simba.mysqlodbc.ini`.

Configuring ODBC Connections on a Non-Windows Machine

The following sections describe how to configure ODBC connections when using the Simba MySQL ODBC Driver on non-Windows platforms:

- [Creating a Data Source Name on a Non-Windows Machine](#) on page 26
- [Configuring a DSN-less Connection on a Non-Windows Machine](#) on page 29
- [Configuring SSL Connections on a Non-Windows Machine](#) on page 31
- [Configuring Logging Options on a Non-Windows Machine](#) on page 33
- [Testing the Connection on a Non-Windows Machine](#) on page 35

Creating a Data Source Name on a Non-Windows Machine

When connecting to your data store using a DSN, you only need to configure the `odbc.ini` file. Set the properties in the `odbc.ini` file to create a DSN that specifies the connection information for your data store. For information about configuring a DSN-less connection instead, see [Configuring a DSN-less Connection on a Non-Windows Machine](#) on page 29.

If your machine is already configured to use an existing `odbc.ini` file, then update that file by adding the settings described below. Otherwise, copy the `odbc.ini` file from the `Setup` subfolder in the driver installation directory to the home directory, and then update the file as described below.

To create a Data Source Name on a non-Windows machine:

1. In a text editor, open the `odbc.ini` configuration file.

 **Note:**

If you are using a hidden copy of the `odbc.ini` file, you can remove the period (.) from the start of the file name to make the file visible while you are editing it.

2. In the `[ODBC Data Sources]` section, add a new entry by typing a name for the DSN, an equal sign (=), and then the name of the driver.

For example, on a macOS machine:

```
[ODBC Data Sources]
```

```
Sample DSN=Simba MySQL ODBC Driver
```

As another example, for a 32-bit driver on a Linux machine:

```
[ODBC Data Sources]
Sample DSN=Simba MySQL ODBC Driver 32-bit
```

3. Create a section that has the same name as your DSN, and then specify configuration options as key-value pairs in the section:
 - a. Set the `Driver` property to the full path of the driver library file that matches the bitness of the application.

For example, on a macOS machine:

```
Driver=/Library/simba/mysql/lib/libmysqlodbc_
sbu.dylib
```

As another example, for a 32-bit driver on a Linux machine:

```
Driver=/opt/simba/mysql/lib/32/libmysqlodbc_sb32.so
```

- b. Set the `Server` property to the IP address or host name of the server, and then set the `Port` property to the number of the TCP port that the server uses to listen for client connections.

For example:

```
Server=192.168.222.160
Port=3306
```

- c. Set the `UID` property to an appropriate user name for accessing the MySQL database.

For example:

```
UID=simba
```

- d. If you are required to authenticate the connection, then set the `PWD` property to the password corresponding to the user name you specified above.

For example:

```
PWD=simba123
```

- e. Optionally, if you are authenticating the connection using the SHA-256 authentication plugin, specify an RSA public key for encrypting your password by setting the `RSAPublicKey` property to the full path and name of the file containing the key.

For example:

```
RSAKey=/localhome/simba/authentication/mysql_rsa.pem
```

 **Note:**

If you do not specify an RSA public key or configure SSL for your connection, the driver automatically retrieves a public key from the server and uses it to encrypt your password.

- f. If you want to enable encryption and identity verification using SSL, then enable SSL and specify the certificate information. For more information, see [Configuring SSL Connections on a Non-Windows Machine](#) on page 31.
 - g. Optionally, set additional key-value pairs as needed to specify other optional connection settings. For detailed information about all the configuration options supported by the Simba MySQL ODBC Driver, see [Driver Configuration Options](#) on page 43.
4. Save the `odbc.ini` configuration file.

 **Note:**

If you are storing this file in its default location in the home directory, then prefix the file name with a period (.) so that the file becomes hidden. If you are storing this file in another location, then save it as a non-hidden file (without the prefix), and make sure that the `ODBCINI` environment variable specifies the location. For more information, see [Specifying the Locations of the Driver Configuration Files](#) on page 24.

For example, the following is an `odbc.ini` configuration file for macOS containing a DSN that connects to MySQL and authenticates the connection using the native plugin:

```
[ODBC Data Sources]
Sample DSN=Simba MySQL ODBC Driver
[Sample DSN]
Driver=/Library/simba/mysql/lib/libmysqlodbc_sbu.dylib
Server=192.168.222.160
Port=3306
UID=simba
PWD=simba123
```

As another example, the following is an `odbc.ini` configuration file for a 32-bit driver on a Linux machine, containing a DSN that connects to MySQL and authenticates the connection using the native plugin:

```
[ODBC Data Sources]
Sample DSN=Simba MySQL ODBC Driver 32-bit
[Sample DSN]
Driver=/opt/simba/mysql/lib/32/libmysqlodbc_sb32.so
Server=192.168.222.160
Port=3306
UID=simba
PWD=simba123
```

You can now use the DSN in an application to connect to the data store.

Configuring a DSN-less Connection on a Non-Windows Machine

To connect to your data store through a DSN-less connection, you need to define the driver in the `odbcinst.ini` file and then provide a DSN-less connection string in your application.

If your machine is already configured to use an existing `odbcinst.ini` file, then update that file by adding the settings described below. Otherwise, copy the `odbcinst.ini` file from the `Setup` subfolder in the driver installation directory to the home directory, and then update the file as described below.

To define a driver on a non-Windows machine:

1. In a text editor, open the `odbcinst.ini` configuration file.

 **Note:**

If you are using a hidden copy of the `odbcinst.ini` file, you can remove the period (.) from the start of the file name to make the file visible while you are editing it.

2. In the `[ODBC Drivers]` section, add a new entry by typing a name for the driver, an equal sign (=), and then `Installed`.

For example:

```
[ODBC Drivers]
Simba MySQL ODBC Driver=Installed
```

3. Create a section that has the same name as the driver (as specified in the previous step), and then specify the following configuration options as key-value pairs in the section:

- a. Set the `Driver` property to the full path of the driver library file that matches the bitness of the application.

For example, on a macOS machine:

```
Driver=/Library/simba/mysql/lib/libmysqlodbc_
sbu.dylib
```

As another example, for a 32-bit driver on a Linux machine:

```
Driver=/opt/simba/mysql/lib/32/libmysqlodbc_sb32.so
```

- b. Optionally, set the `Description` property to a description of the driver.

For example:

```
Description=Simba MySQL ODBC Driver
```

4. Save the `odbcinst.ini` configuration file.

 **Note:**

If you are storing this file in its default location in the home directory, then prefix the file name with a period (.) so that the file becomes hidden. If you are storing this file in another location, then save it as a non-hidden file (without the prefix), and make sure that the `ODBCINSTINI` or `ODBCSYSINI` environment variable specifies the location. For more information, see [Specifying the Locations of the Driver Configuration Files](#) on page 24.

For example, the following is an `odbcinst.ini` configuration file for macOS:

```
[ODBC Drivers]
Simba MySQL ODBC Driver=Installed
[Simba MySQL ODBC Driver]
Description=Simba MySQL ODBC Driver
Driver=/Library/simba/mysql/lib/libmysqlodbc_sbu.dylib
```

As another example, the following is an `odbcinst.ini` configuration file for both the 32- and 64-bit drivers on Linux:

```
[ODBC Drivers]
Simba MySQL ODBC Driver 32-bit=Installed
Simba MySQL ODBC Driver 64-bit=Installed
[Simba MySQL ODBC Driver 32-bit]
Description=Simba MySQL ODBC Driver (32-bit)
Driver=/opt/simba/mysql/lib/32/libmysqlodbc_sb32.so
```

```
[Simba MySQL ODBC Driver 64-bit]
Description=Simba MySQL ODBC Driver (64-bit)
Driver=/opt/simba/mysql/lib/64/libmysqlodbc_sb64.so
```

You can now connect to your data store by providing your application with a connection string where the `Driver` property is set to the driver name specified in the `odbcinst.ini` file, and all the other necessary connection properties are also set. For more information, see "DSN-less Connection String Examples" in [Using a Connection String](#) on page 37.

For instructions about configuring SSL connections, see [Configuring SSL Connections on a Non-Windows Machine](#) on page 31.

For detailed information about all the connection properties that the driver supports, see [Driver Configuration Options](#) on page 43.

Configuring SSL Connections on a Non-Windows Machine

If you are connecting to a MySQL server that has Secure Sockets Layer (SSL) enabled, then you can configure the driver to connect to an SSL-enabled socket and encrypt the connection. When connecting to a server over SSL, the driver supports identity verification between the client and the server.

You can set the connection properties described below in a connection string or in a DSN (in the `odbc.ini` file). Settings in the connection string take precedence over settings in the DSN.

Configuring an SSL Connection without Identity Verification

You can configure a connection that is encrypted by SSL but does not verify the identity of the client or the server.

To configure an SSL connection without verification on a non-Windows machine:

1. Enable SSL encryption by doing one of the following:
 - To use SSL encryption only if the server supports it, set the `SSLMode` property to 1.
 - Or, to require SSL encryption for the connection, set the `SSLMode` property to 2. If the server does not support SSL, the connection fails.
2. Optionally, set the `SSLCipher` property to a comma-separated list of permitted ciphers for encrypting the connection.

3. To specify the minimum version of SSL to use, set the `Min_TLS` property to the minimum version of SSL. Supported options include `1.0` for TLS 1.0, `1.1` for TLS 1.1, and `1.2` for TLS 1.2.

Configuring SSL Identity Verification

You can configure one-way verification so that the client verifies the identity of the MySQL server, or you can configure two-way verification so that the client and the sever both verify each other.

In both cases, you must provide a root certificate from a trusted certificate authority (CA) that the driver can use to check the server's certificate. If you are using two-way verification, then you must also provide a certificate that proves the identity of the client and a private key that encrypts the client certificate.

To configure SSL identity verification on a non-Windows machine:

1. Enable SSL encryption by doing one of the following:
 - To use SSL encryption and identity verification only if the server supports it, set the `SSLMode` property to `3`.
 - Or, to require SSL encryption and identity verification for the connection, set the `SSLMode` property to `4`. If the server does not support SSL or if identity verification fails, the connection fails.
2. To specify one or more root certificates from trusted CAs that you want to use to verify the server certificate, do one of the following:
 - To use a specific root certificate, set the `SSLCA` property to the full path and name of the `.pem` file containing the certificate.
 - Or, to provide multiple root certificates, set the `SSLCAPath` property to the full path and name of the directory that contains the certificates. The driver uses the first valid certificate that it finds in the directory.
3. If two-way identity verification is necessary, do the following:
 - a. Set the `SSLCert` property to the full path and name of the `.pem` file containing the certificate used for proving the identity of the client.
 - b. Set the `SSLKey` property to the full path and name of the file that contains the private key used for encrypting the client certificate.
4. Optionally, set the `SSLCipher` property to a comma-separated list of permitted ciphers for encrypting the connection.
5. To specify the minimum version of SSL to use, set the `Min_TLS` property to the minimum version of SSL. Supported options include `1.0` for TLS 1.0, `1.1` for TLS 1.1, and `1.2` for TLS 1.2.

Configuring Logging Options on a Non-Windows Machine

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.

Logging is configured through driver-wide settings in the `simba.mysqlodbc.ini` file, which apply to all connections that use the driver.

To enable logging on a non-Windows machine:

1. Open the `simba.mysqlodbc.ini` configuration file in a text editor.
2. To specify the level of information to include in log files, set the `LogLevel` property to one of the following numbers:

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

3. Set the `LogPath` key to the full path to the folder where you want to save log files.
4. Set the `LogFileCount` key to the maximum number of log files to keep.

 **Note:**

After the maximum number of log files is reached, each time an additional file is created, the driver deletes the oldest log file.

5. Set the `LogFileSize` key to the maximum size of each log file in megabytes (MB).

 **Note:**

After the maximum file size is reached, the driver creates a new file and continues logging.

6. Optionally, to prefix the log file name with the user name and process ID associated with the connection, set the `UseLogPrefix` property to 1.
7. Save the `simba.mysqlodbc.ini` configuration file.
8. Restart your ODBC application to make sure that the new settings take effect.
 - A `[DriverName]_connection_[Number].log` for each connection made to the database, where `[Number]` is a number that identifies each log file. This file logs driver activity that is specific to the connection.

The Simba MySQL ODBC Driver produces the following log files at the location you specify using the `LogPath` key:

- A `simbamysqlodbcdriver.log` file that logs driver activity that is not specific to a connection.
- A `simbamysqlodbcdriver_connection_[Number].log` file for each connection made to the database, where `[Number]` is a number that identifies each log file. This file logs driver activity that is specific to the connection.

If you set the `UseLogPrefix` property to 1, then each file name is prefixed with `[UserName]_[ProcessID]_`, where `[UserName]` is the user name associated with the connection and `[ProcessID]` is the process ID of the application through which the connection is made. For more information, see [UseLogPrefix](#) on page 61.

To disable logging on a non-Windows machine:

1. Open the `simba.mysqlodbc.ini` configuration file in a text editor.
2. Set the `LogLevel` key to 0.
3. Save the `simba.mysqlodbc.ini` configuration file.
4. Restart your ODBC application to make sure that the new settings take effect.

Testing the Connection on a Non-Windows Machine

To test the connection, you can use an ODBC-enabled client application. For a basic connection test, you can also use the test utilities that are packaged with your driver manager installation. For example, the iODBC driver manager includes simple utilities called `iodbctest` and `iodbctestw`. Similarly, the unixODBC driver manager includes simple utilities called `isql` and `iusql`.

Using the iODBC Driver Manager

You can use the `iodbctest` and `iodbctestw` utilities to establish a test connection with your driver. Use `iodbctest` to test how your driver works with an ANSI application, or use `iodbctestw` to test how your driver works with a Unicode application.

 **Note:**

There are 32-bit and 64-bit installations of the iODBC driver manager available. If you have only one or the other installed, then the appropriate version of `iodbctest` (or `iodbctestw`) is available. However, if you have both 32- and 64-bit versions installed, then you need to make sure that you are running the version from the correct installation directory.

For more information about using the iODBC driver manager, see <http://www.iodbc.org>.

To test your connection using the iODBC driver manager:

1. Run **`iodbctest`** or **`iodbctestw`**.
2. Optionally, if you do not remember the DSN, then type a question mark (?) to see a list of available DSNs.
3. Type the connection string for connecting to your data store, and then press ENTER. For more information, see [Using a Connection String](#) on page 37 .

If the connection is successful, then the `SQL>` prompt appears.

Using the unixODBC Driver Manager

You can use the `isql` and `iusql` utilities to establish a test connection with your driver and your DSN. `isql` and `iusql` can only be used to test connections that use a DSN. Use `isql` to test how your driver works with an ANSI application, or use `iusql` to test how your driver works with a Unicode application.

 **Note:**

There are 32-bit and 64-bit installations of the unixODBC driver manager available. If you have only one or the other installed, then the appropriate version of isql (or iusql) is available. However, if you have both 32- and 64-bit versions installed, then you need to make sure that you are running the version from the correct installation directory.

For more information about using the unixODBC driver manager, see <http://www.unixodbc.org>.

To test your connection using the unixODBC driver manager:

- Run isql or iusql by using the corresponding syntax:

- `isql [DataSourceName]`
- `iusql [DataSourceName]`

`[DataSourceName]` is the DSN that you are using for the connection.

If the connection is successful, then the `SQL>` prompt appears.

 **Note:**

For information about the available options, run isql or iusql without providing a DSN.

Using a Connection String

For some applications, you might need to use a connection string to connect to your data source. For detailed information about how to use a connection string in an ODBC application, refer to the documentation for the application that you are using.

The connection strings in the following sections are examples showing the minimum set of connection attributes that you must specify to successfully connect to the data source. Depending on the configuration of the data source and the type of connection you are working with, you might need to specify additional connection attributes. For detailed information about all the attributes that you can use in the connection string, see [Driver Configuration Options](#) on page 43.

DSN Connection String Example

The following is an example of a connection string for a connection that uses a DSN:

```
DSN=[DataSourceName]
```

[DataSourceName] is the DSN that you are using for the connection.

You can set additional configuration options by appending key-value pairs to the connection string. Configuration options that are passed in using a connection string take precedence over configuration options that are set in the DSN.

DSN-less Connection String Examples

Some applications provide support for connecting to a data source using a driver without a DSN. To connect to a data source without using a DSN, use a connection string instead.

! Important:

When you connect to the data store using a DSN-less connection string, the driver does not encrypt your credentials.

The placeholders in the examples are defined as follows, in alphabetical order:

- *[PortNumber]* is the number of the TCP port that the MySQL server uses to listen for client connections.
- *[ServerInfo]* is the IP address or host name of the MySQL server to which you are connecting.

- *[YourPassword]* is the password corresponding to your user name.
- *[YourUserName]* is the user name that you use to access the MySQL server.

Connecting to a MySQL Server that Uses Windows Native Authentication or No Authentication

The following is the format of a DSN-less connection string for connecting to a MySQL server that either uses the Windows native authentication plugin or does not require authentication. You must always specify a user name when connecting to MySQL, so this same connection string format can be used for both types of connections.

```
Driver=Simba MySQL ODBC Driver;Server=[ServerInfo];  
Port=[PortNumber];UID=[YourUserName]
```

For example:

```
Driver=Simba MySQL ODBC Driver;Server=192.168.222.160;  
Port=3306;UID=simba
```

Connecting to a MySQL Server that Uses Native or SHA-256 Authentication

The following is the format of a DSN-less connection string for connecting to a MySQL server that uses either the native or SHA-256 authentication plugin:

```
Driver=Simba MySQL ODBC Driver;Server=[ServerInfo];  
Port=[PortNumber];UID=[YourUserName];PWD=[YourPassword]
```

For example:

```
Driver=Simba MySQL ODBC Driver;Server=192.168.222.160;  
Port=3306;UID=simba;PWD=simba123
```

Features

For more information on the features of the Simba MySQL ODBC Driver, see the following:

- [Data Types](#) on page 39
- [Security and Authentication](#) on page 42

Data Types

The Simba MySQL ODBC Driver supports many common data formats, converting between MySQL data types and SQL data types.

The table below lists the supported data type mappings.

MySQL Type	SQL Type
BIGINT	SQL_BIGINT <div style="border: 1px solid black; padding: 5px; background-color: #f0f0f0;"> <p> Note: SQL_INTEGER is returned instead if the Treat BIGINT Columns As INT Columns option (the No_BIGINT property) is enabled.</p> </div>
BINARY	SQL_BINARY
BIT (M)	<ul style="list-style-type: none"> • SQL_BIT when M = 1 • SQL_BINARY when M > 1
BLOB	SQL_LONGVARBINARY
BOOL BOOLEAN	SQL_TINYINT
CHAR	SQL_WCHAR

MySQL Type	SQL Type
DATE	<ul style="list-style-type: none"> • SQL_TYPE_DATE if the application uses ODBC version 3.00 or later. • SQL_DATE if the application uses an ODBC version earlier than 3.00.
DATETIME	<ul style="list-style-type: none"> • SQL_TYPE_TIMESTAMP if the application uses ODBC version 3.00 or later. • SQL_TIMESTAMP if the application uses an ODBC version earlier than 3.00.
DEC DECIMAL	SQL_DECIMAL
DOUBLE DOUBLE PRECISION	SQL_DOUBLE
ENUM	SQL_WCHAR
FLOAT	SQL_REAL
INT INTEGER	SQL_INTEGER
JSON	SQL_BINARY
 Note: Only supported in MySQL 5.7 or later.	
LONGBLOB	SQL_LONGVARBINARY
LONGTEXT	SQL_WLONGVARCHAR

MySQL Type	SQL Type
MEDIUMBLOB	SQL_LONGVARBINARY
MEDIUMINT	SQL_INTEGER
MEDIUMTEXT	SQL_WLONGVARCHAR
NUMERIC	SQL_DECIMAL
SET	SQL_WCHAR
SMALLINT	SQL_SMALLINT
TEXT	SQL_WLONGVARCHAR
TIME	<ul style="list-style-type: none"> • SQL_TYPE_TIME if the application uses ODBC version 3.00 or later. • SQL_TIME if the application uses an ODBC version earlier than 3.00.
TIMESTAMP	<ul style="list-style-type: none"> • SQL_TYPE_TIMESTAMP if the application uses ODBC version 3.00 or later. • SQL_TIMESTAMP if the application uses an ODBC version earlier than 3.00.
TINYBLOB	SQL_LONGVARBINARY
TINYINT	SQL_TINYINT
TINYTEXT	SQL_WLONGVARCHAR
VARBINARY	SQL_VARBINARY
VARCHAR	SQL_WVARCHAR
YEAR	SQL_SMALLINT

Security and Authentication

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

 **Note:**

In this documentation, "SSL" refers to both TLS (Transport Layer Security) and SSL (Secure Sockets Layer). The SSL version used for the connection is the highest version that is supported by both the driver and the server.

By default, the driver supports TLS 1.0, 1.1, and 1.2. You can configure the driver to use a specific TLS version. For more information, see [Configuring SSL Connections on Windows](#) on page 14 or [Configuring SSL Connections on a Non-Windows Machine](#) on page 31.

The driver provides a mechanism that enables you to authenticate your connection using the native, SHA-256, or Windows native authentication plugins. The settings in the MySQL server determine which plugin is used and what credentials you need to provide. For detailed driver configuration instructions, see [Creating a Data Source Name on Windows](#) on page 9 or [Creating a Data Source Name on a Non-Windows Machine](#) on page 26.

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

- No identity verification
- 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 Windows](#) on page 14 or [Configuring SSL Connections on a Non-Windows Machine](#) on page 31.

Driver Configuration Options

Driver Configuration Options lists the configuration options available in the Simba MySQL ODBC Driver alphabetically by field or button label. Options having only key names, that is, not appearing in the user interface of the driver, are listed alphabetically by key name.

When creating or configuring a connection from a Windows machine, the fields and buttons described below are available in the following dialog boxes:

- Simba MySQL ODBC Driver DSN Setup
- Logging Options

When using a connection string or configuring a connection from a non-Windows machine, use the key names provided below.

Configuration Options Appearing in the User Interface

The following configuration options are accessible via the Windows user interface for the Simba MySQL ODBC Driver, or via the key name when using a connection string or configuring a connection from a Linux or macOS machine:

- [Allow Multiple Statements](#) on page 44
- [Always Handle Binary Function Results As Character Data](#) on page 44
- [Bind Minimal Date As Zero Date](#) on page 45
- [Can Handle Expired Password](#) on page 45
- [Catalog](#) on page 46
- [Disable Transaction Support](#) on page 46
- [Don't Cache Results Of Forward-Only Cursors](#) on page 46
- [Don't Prompt When Connecting](#) on page 47
- [Enable Automatic Reconnect](#) on
- [Max Number Files](#) on page 53
- [Minimum TLS](#) on page 53
- [Password](#) on page 53
- [Port](#) on page 54
- [Prepare Statements On The Client](#) on page 54
- [Return Matched Rows Instead Of Affected Rows](#) on page 54
- [Return Minimal Date For Zero Date](#) on page 55
- [RSA Public Key](#) on page 55
- [SSL CA Path](#) on page 56
- [SSL Certificate](#) on page 56
- [SSL Certificate Authority](#) on page 57
- [SSL Cipher](#) on page 57

- page 47
- [Enable SQL_AUTO_IS_NULL](#) on page 47
- [Enable Table Types](#) on page 48
- [Host](#) on page 48
- [Ignore Schema In Column Specifications](#) on page 49
- [Ignore Space After Function Names](#) on page 49
- [Include Table Name In SQLDescribeCol\(\)](#) on page 49
- [Initial Statement](#) on page 50
- [Interactive Client](#) on page 50
- [Log Level](#) on page 50
- [Log Path](#) on page 51
- [Limit column size to a signed 32-bit value](#) on page 52
- [Max File Size](#) on page 52
- [SSL Key](#) on page 57
- [SSL Mode](#) on page 58
- [Treat BIGINT Columns As INT Columns](#) on page 58
- [TrustedCertificates](#) on page 59
- [Use Compression](#) on page 59
- [Use Trust Store](#) on page 60
- [User](#) on page 60

Allow Multiple Statements

Key Name	Default Value	Required
Multi_Statements	Clear (0)	No

Description

This option specifies whether the driver supports batched statements.

- Enabled (1): The driver allows multiple statements to be processed in a single execution.
- Disabled (0): The driver does not allow batched statements.

Always Handle Binary Function Results As Character Data

Key Name	Default Value	Required
No_Binary_Result	Clear (0)	No

Description

This option specifies whether the driver returns Binary columns as Character columns when returning function results.

- Enabled (1): The driver returns Binary columns as Character columns.
- Disabled (0): The driver returns Binary columns normally.

Bind Minimal Date As Zero Date

Key Name	Default Value	Required
Min_Date_To_Zero	Clear (0)	No

Description

This option specifies whether the driver translates the minimum ODBC date value (0000-01-01) to the zero date value used in MySQL (0000-00-00) when binding parameters. Enabling this option prevents statements from failing because of what seems to be a mismatch in the date values.

- Enabled (1): The driver translates the minimum ODBC date value to the MySQL zero date value when binding parameters.
- Disabled (0): The driver does not modify the date values.

Can Handle Expired Password

Key Name	Default Value	Required
Can_Handle_Exp_Pwd	Clear (0)	No

Description

This option determines how the driver responds when you attempt to connect to the server using an expired password.

- Enabled (1): The driver establishes a sandboxed connection through which you can issue a SET PASSWORD statement to update the password.
- Disabled (0): The driver returns an error, and the connection fails.

Catalog

Key Name	Default Value	Required
Database	None	No

Description

The name of the MySQL database to connect to by default. You can still issue queries on other databases by specifying the database schema in your query statement.

Disable Transaction Support

Key Name	Default Value	Required
No_Transactions	Clear (0)	No

Description

This option specifies whether the driver supports transactions.

- Enabled (1): The driver does not support transactions.
- Disabled (0): The driver supports transactions.

Don't Cache Results Of Forward-Only Cursors

Key Name	Default Value	Required
No_Cache	Clear (0)	No

Description

This option specifies whether the driver caches result sets into memory. Enable this option to reduce memory consumption when working with large result sets.

- Enabled (1): The driver does not cache result sets into memory.
- Disabled (0): The driver caches result sets into memory.

Don't Prompt When Connecting

Key Name	Default Value	Required
No_Prompt	Clear (0)	No

Description

This option specifies whether the driver allows dialog boxes to open and prompt you for more information during connection time.

- Enabled (1): The driver does not allow dialog boxes to open.
- Disabled (0): The driver allows dialog boxes to open and prompt you for more connection information, if necessary.

Enable Automatic Reconnect

Key Name	Default Value	Required
Auto_Reconnect	Clear (0)	No

Description

This option specifies whether the driver attempts to automatically reconnect to the server when a communication link error occurs.

! Important:

Do not enable this option if you are working with transactions. Auto-reconnecting during an incomplete transaction may corrupt the data.

- Enabled (1): The driver attempts to reconnect.
- Disabled (0): The driver does not attempt to reconnect.

Enable SQL_AUTO_IS_NULL

Key Name	Default Value	Required
Auto_Is_Null	Clear (0)	No

Description

This option specifies whether the driver modifies the `sql_auto_is_null` setting in the server. For more information, see "Server System Variables" in the *MySQL Reference Manual*: https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html#sysvar_sql_auto_is_null.

- Enabled (1): The driver does not modify the `sql_auto_is_null` setting in the server, and supports `sql_auto_is_null=1` if it is specified in the server.
- Disabled (0): The driver sets the `sql_auto_is_null` server system variable to 0, and does not support the retrieval of `AUTO_INCREMENT` values using IS NULL queries.

Enable Table Types

Key Name	Default Value	Required
<code>EnableTableTypes</code>	Clear (0)	No

Description

This option specifies whether the driver recognizes table type information from the data source. By default, the driver only recognizes a single, generic table type.

- Enabled (1): The driver recognizes the following table types: TABLE and SYSTEM TABLE.
- Disabled (0): All tables returned from the data source have the generic type TABLE.

Host

Key Name	Default Value	Required
<code>Server</code>	<code>localhost</code>	Yes

Description

The host name or IP address of the MySQL server.

Ignore Schema In Column Specifications

Key Name	Default Value	Required
No_Schema	Clear (0)	No

Description

This option specifies whether the driver accepts column names that include the database name (using the format *[Database].[Table].[Column]*).

- Enabled (1): The driver ignores the database name.
- Disabled (0): The driver returns an error if a column name specified in a statement includes the database name.

Ignore Space After Function Names

Key Name	Default Value	Required
Ignore_Space	Clear (0)	No

Description

This option specifies whether the driver ignores spaces between function names and parentheses () so that function names are treated as keywords. Enabling this option provides compatibility with applications such as PowerBuilder.

- Enabled (1): The driver ignores the spaces, and treats function names as keywords.
- Disabled (0): The driver does not ignore the spaces.

Include Table Name In SQLDescribeCol()

Key Name	Default Value	Required
Full_Column_Names	Clear (0)	No

Description

This option specifies whether the driver returns fully-qualified column names when the `SQLDescribeCol()` function is called.

- Enabled (1): The driver returns fully-qualified column names.
- Disabled (0): The driver returns column names that do not include the table name.

Initial Statement

Key Name	Default Value	Required
InitStmt	None	No

Description

A statement that the driver executes immediately after connecting to the server.

Interactive Client

Key Name	Default Value	Required
Interactive	Clear (0)	No

Description

This option specifies whether the driver manages the connection using the `interactive_timeout` value specified in the server.

- Enabled (1): The driver closes the connection if the time interval specified by `interactive_timeout` passes without any activity occurring.
- Disabled (0): The driver does not close inactive connections.

Log Level

Key Name	Default Value	Required
LogLevel	OFF (0)	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 MySQL ODBC Driver, so make sure to disable the feature after you are done using it.
- This option is not supported in connection strings. To configure logging for the Windows driver, you must use the Logging Options dialog box. To configure logging for a non-Windows driver, you must use the `simba.mysqlodbc.ini` file.

Set the property to one of the following values:

- OFF (0): Disable all logging.
- FATAL (1): Logs severe error events that lead the driver to abort.
- ERROR (2): Logs error events that might allow the driver to continue running.
- WARNING (3): Logs events that might result in an error if action is not taken.
- INFO (4): Logs general information that describes the progress of the driver.
- DEBUG (5): Logs detailed information that is useful for debugging the driver.
- TRACE (6): Logs all driver activity.

When logging is enabled, the driver produces the following log files at the location you specify in the Log Path (`LogPath`) property:

- A `simbamysqlodbcdriver.log` file that logs driver activity that is not specific to a connection.
- A `simbamysqlodbcdriver_connection_[Number].log` file for each connection made to the database, where `[Number]` is a number that identifies each log file. This file logs driver activity that is specific to the connection.

If you enable the `UseLogPrefix` connection property, the driver prefixes the log file name with the user name associated with the connection and the process ID of the application through which the connection is made. For more information, see [UseLogPrefix](#) on page 61.

Log Path

Key Name	Default Value	Required
<code>LogPath</code>	None	Yes, if logging is enabled.

Description

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

! Important:

This option is not supported in connection strings. To configure logging for the Windows driver, you must use the Logging Options dialog box. To configure logging for a non-Windows driver, you must use the `simba.mysqlodbc.ini` file.

Limit column size to a signed 32-bit value

Key Name	Default Value	Required
COLUMN_SIZE_S32	Clear (0)	No

Description

This option specifies whether the driver limits column sizes to 32-bit values, preventing problems with larger column sizes in applications that do not support them.

- 0: The driver uses 32-bit and 64-bit column sizes as required by the data.
- 1: The driver limits column sizes to signed 32-bit values.

Max File Size

Key Name	Default Value	Required
LogFileSize	20	No

Description

The maximum size of each log file in megabytes (MB). After the maximum file size is reached, the driver creates a new file and continues logging.

! Important:

This option is not supported in connection strings. To configure logging for the Windows driver, you must use the Logging Options dialog box. To configure logging for a non-Windows driver, you must use the `simba.mysqlodbc.ini` file.

Max Number Files

Key Name	Default Value	Required
LogFileCount	50	No

Description

The maximum number of log files to keep. After the maximum number of log files is reached, each time an additional file is created, the driver deletes the oldest log file.

! Important:

This option is not supported in connection strings. To configure logging for the Windows driver, you must use the Logging Options dialog box. To configure logging for a non-Windows driver, you must use the `simba.mysqlodbc.ini` file.

Minimum TLS

Key Name	Default Value	Required
Min_TLS	TLS 1.2 (1.2)	No

Description

The minimum version of TLS/SSL that the driver allows the data store to use for encrypting connections. For example, if TLS 1.1 is specified, TLS 1.0 cannot be used to encrypt connections.

- TLS 1.0 (1.0): The connection must use at least TLS 1.0.
- TLS 1.1 (1.1): The connection must use at least TLS 1.1.
- TLS 1.2 (1.2): The connection must use at least TLS 1.2.

Password

Key Name	Default Value	Required
PWD OR Password	None	Yes, if connecting to a server that uses the Native or SHA-256 authentication plugins.

Description

The password corresponding to the user name that you provided in the User field (the `User` or `UID` key).

Port

Key Name	Default Value	Required
Port	3306	Yes

Description

The TCP port that the MySQL server uses to listen for client connections.

Prepare Statements On The Client

Key Name	Default Value	Required
No_SSPS	Clear (0)	No

Description

This option specifies whether the driver emulates prepared statements on the client side. This functionality is required when using prepared statements for certain types of queries. If the driver fails to execute your prepared statement, enable this option and then try executing the statement again.

- Enabled (1): The driver emulates prepared statements on the client side.
- Disabled (0): The driver does not emulate prepared statements on the client side.

Return Matched Rows Instead Of Affected Rows

Key Name	Default Value	Required
Found_Rows	Clear (0)	No

Description

This option specifies whether the driver returns the number of matched rows or the number of affected rows after a DML operation.

- Enabled (1): The driver returns the number of matched rows.
- Disabled (0): The driver returns the number of affected rows.

Return Minimal Date For Zero Date

Key Name	Default Value	Required
ZeroDateToMin	0	No

Description

This option specifies whether the driver translates zero dates (where the month or day is 0) to a date value that is supported by ODBC. For example, when this option is enabled, the date (0000-00-00) is translated to (0000-01-01).

- Enabled (1): The driver translates the zero dates to a date value that is supported by ODBC.
- Disabled (0): The driver returns zero dates as NULL.

RSA Public Key

Key Name	Default Value	Required
RSAPublicKey	None	No

Description

The full path and name of a file containing an RSA public key for encrypting your password.

This option is applicable only when you connect to a server that uses the SHA-256 authentication plugin.

SSL CA Path

Key Name	Default Value	Required
SSLCAPath	None	Yes, if using SSL identity verification and the SSL Certificate Authority option (the <code>SSLCA</code> property) is not set.

Description

The full path and name of the directory that contains root certificates from trusted CAs. The certificates must be stored as `.pem` files.

Use this option to specify root certificates for verifying the server certificate during an SSL identity verification. The driver uses the first valid certificate that it finds in the directory.

To use a specific root certificate, configure the SSL Certificate Authority option (the `SSLCA` property) instead. For more information, see [SSL Certificate Authority](#) on page 57.

SSL Certificate

Key Name	Default Value	Required
SSLCert	None	Yes, if using two-way SSL verification.

Description

The full path and name of a `.pem` file containing the SSL certificate used for proving the identity of the client.

To specify a private key for encrypting this certificate before sending it to the server, use the SSL Key option (the `SSLKey` property). For more information, see [SSL Key](#) on page 57.

SSL Certificate Authority

Key Name	Default Value	Required
SSLCA	None	Yes, if using SSL identity verification and the SSL CA Path option (the SSLCAPath property) is not set.

Description

The full path and name of a .pem file containing a root certificate from a trusted CA.

Use this option to specify a root certificate for verifying the server certificate during SSL identity verification.

SSL Cipher

Key Name	Default Value	Required
SSLCipher	None	No

Description

A comma-separated list of permitted ciphers for the SSL connection.

If you do not set this option, the driver automatically selects an appropriate cipher for the connection.

SSL Key

Key Name	Default Value	Required
SSLKey	None	Yes, if using two-way SSL verification.

Description

The full path and name of a file containing the private key used for encrypting the client-side certificate during two-way SSL verification.

The client-side certificate is specified using the SSL Certificate option (the `SSLCert` property). For more information, see [SSL Certificate](#) on page 56.

SSL Mode

Key Name	Default Value	Required
<code>SSLMode</code>	PREFERRED (1)	No

Description

This option specifies whether the driver uses SSL encryption and verification when connecting to MySQL.

- **DISABLED** (0): SSL is not used.
- **PREFERRED** (1): Use SSL encryption if the server supports it. Otherwise, connect without using SSL.
- **REQUIRED** (2): Use SSL encryption. If the server does not support SSL, the connection fails.
- **VERIFY_CA** (3): Use SSL encryption and either one-way or two-way identity verification. If the server does not support SSL or if identity verification fails, the connection fails.
- **VERIFY_IDENTITY** (4): Use SSL encryption and either one-way or two-way identity verification, and also verify the host name of the server. If the server does not support SSL or if a certificate or host name cannot be verified, the connection fails.

Treat BIGINT Columns As INT Columns

Key Name	Default Value	Required
<code>NO_BIGINT</code>	Clear (0)	No

Description

This option specifies whether the driver returns BIGINT data as `SQL_BIGINT` or `SQL_INTEGER` data.

- Enabled (1): The driver returns BIGINT data as `SQL_INTEGER` data.
- Disabled (0): The driver returns BIGINT data as `SQL_BIGINT` data.

TrustedCertificates

Key Name	Default Value	Required
TrustedCerts	The <code>cacerts.pem</code> file in the <code>\lib</code> subfolder within the driver's installation directory. The exact file path varies depending on the version of the driver that is installed. For example, the path for the Windows driver is different from the path for the macOS driver.	No

Description

The full path of the `.pem` file containing trusted CA certificates, for verifying the server.

If this option is not set, then the driver defaults to using the trusted CA certificates `.pem` file installed by the driver.

Use Compression

Key Name	Default Value	Required
Compressed_Proto	Clear (0)	No

Description

This option specifies whether the driver compresses the communication between the client and the server. Compression may reduce the amount of network traffic for certain workflows, but in some cases it may increase CPU usage.

- Enabled (1): The driver compresses the communication between the client and the server.
- Disabled (0): The driver does not use compression.

Use Trust Store

Key Name	Default Value	Required
UseTrustStore	Clear (0)	No

Description

This option specifies whether to use a CA certificate from the system trust store, or from a specified PEM file.

- Enabled (1): The driver verifies the connection using a certificate in the system trust store.
- Disabled (0): The driver verifies the connection using a specified `.pem` file. For information about specifying a `.pem` file, see [.pem](#).



Note:

This option is only available on Windows.

User

Key Name	Default Value	Required
UID	None	Yes
OR		
User		

Description

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

Configuration Options Having Only Key Names

The `Driver` configuration option does not appear in the Windows user interface for the Simba MySQL ODBC Driver. It is accessible only when you use a connection string or configure a connection on macOS or Linux.

The `UseLogPrefix` property must be configured as a Windows Registry key value, or as a driver-wide property in the `simba.mysqlodbc.ini` file for macOS or Linux.

- [UseLogPrefix](#) on page 61

Driver

Key Name	Default Value	Required
Driver	Simba MySQL ODBC Driver when installed on Windows, or the absolute path of the driver shared object file when installed on a non-Windows machine.	Yes

Description

On Windows, the name of the installed driver (Simba MySQL ODBC Driver).

On other platforms, the name of the installed driver as specified in `odbcinst.ini`, or the absolute path of the driver shared object file.

UseLogPrefix

Key Name	Default Value	Required
UseLogPrefix	0	No

Description

This option specifies whether the driver includes a prefix in the names of log files so that the files can be distinguished by user and application.

! Important:

To configure this option for the Windows driver, you create a value for it in one of the following registry keys:

- For a 32-bit driver installed on a 64-bit machine: `HKEY_LOCAL_MACHINE\SOFTWAREWow6432Node\Simba\Simba MySQL ODBC Driver\Driver`
- Otherwise: `HKEY_LOCAL_MACHINE\SOFTWARE\Simba\Simba MySQL ODBC Driver\Driver`

Use `UseLogPrefix` as the value name, and either `0` or `1` as the value data.

To configure this option for a non-Windows driver, you must use the `simba.mysqlodbc.ini` file.

Set the property to one of the following values:

- `1`: The driver prefixes log file names with the user name and process ID associated with the connection that is being logged.

For example, if you are connecting as a user named "jdoe" and using the driver in an application with process ID 7836, the generated log file would be named `jdoe_7836_SimbaMySQLODBCDriver.log` and `jdoe_7836_SimbaMySQLODBCDriver_connection_[Number].log`, where *[Number]* is a number that identifies each connection-specific log file.

- `0`: The driver does not include the prefix in log file names.

Third-Party Trademarks

Linux is the registered trademark of Linus Torvalds in Canada, United States and/or other countries.

Mac, macOS, Mac OS, and OS X are trademarks or registered trademarks of Apple, Inc. or its subsidiaries in Canada, United States and/or other countries.

Microsoft, MSDN, Windows, Windows Server, Windows Vista, and the Windows start button are trademarks or registered trademarks of Microsoft Corporation or its subsidiaries in Canada, United States and/or other countries.

Red Hat, Red Hat Enterprise Linux, and CentOS are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in Canada, United States and/or other countries.

SUSE is a trademark or registered trademark of SUSE LLC or its subsidiaries in Canada, United States and/or other countries.

MySQL is a registered trademark of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

All other trademarks are trademarks of their respective owners.

Third-Party Licenses

The licenses for the third-party libraries that are included in this product are listed below.

CityHash License

Copyright (c) 2011 Google, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CityHash, by Geoff Pike and Jyrki Alakuijala

<http://code.google.com/p/cityhash/>

dtoa License

The author of this software is David M. Gay.

Copyright (c) 1991, 2000, 2001 by Lucent Technologies.

Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software.

THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHOR NOR LUCENT MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING

THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

Expat License

Copyright (c) 1998, 1999, 2000 Thai Open Source Software Center Ltd

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

ICU License - ICU 1.8.1 and later

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2014 International Business Machines Corporation and others

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES,

OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.

OpenSSL License

Copyright (c) 1998-2016 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment:

"This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"

4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment:

"This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,

INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)

All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following conditions are aheared to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

"This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)"

The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related :-).

4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement:

"This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

Stringencoders License

Copyright 2005, 2006, 2007

Nick Galbreath -- nickg [at] modp [dot] com

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the modp.com nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This is the standard "new" BSD license:

<http://www.opensource.org/licenses/bsd-license.php>

zlib License

Copyright notice:

(C) 1995-2017 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly

Mark Adler

jloup@gzip.org

madler@alumni.caltech.edu