

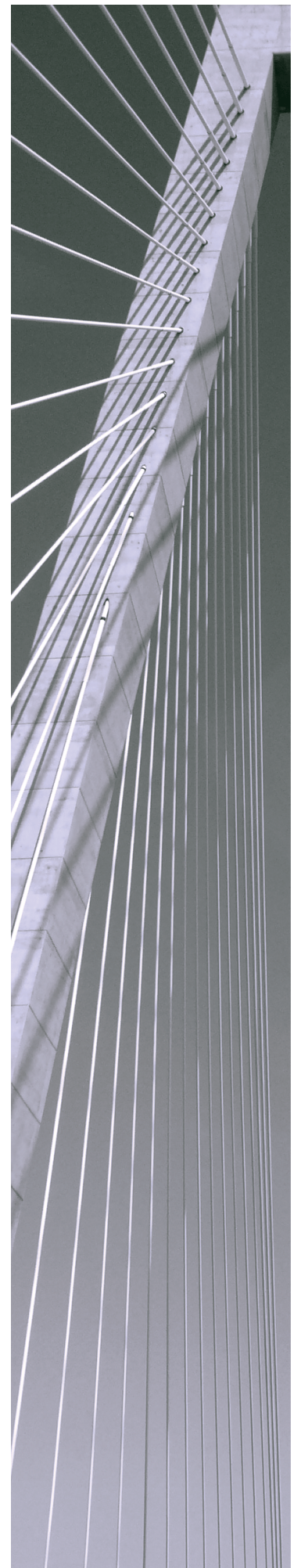


# Simba SQL Server ODBC Driver

## Installation and Configuration Guide

Simba Technologies Inc.

Version 1.3.13  
September 27, 2017



**Copyright © 2017 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](http://www.simba.com)

## About This Guide

### Purpose

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

### Audience

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

### Knowledge Prerequisites

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

- Familiarity with the platform on which you are using the Simba SQL Server ODBC Driver
- Ability to use the data source to which the Simba SQL Server 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 SQL Server 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 Logging Options on Windows .....	11
Verifying the Driver Version Number on Windows .....	15
macOS Driver .....	16
macOS System Requirements .....	16
Installing the Driver on macOS .....	16
Verifying the Driver Version Number on macOS .....	17
Linux Driver .....	18
Linux System Requirements .....	18
Installing the Driver Using the Tarball Package .....	18
Configuring the ODBC Driver Manager on Non-Windows Machines .....	20
Specifying ODBC Driver Managers on Non-Windows Machines .....	20
Specifying the Locations of the Driver Configuration Files .....	21
Configuring ODBC Connections on a Non-Windows Machine .....	23
Creating a Data Source Name on a Non-Windows Machine .....	23
Configuring a DSN-less Connection on a Non-Windows Machine .....	26
Configuring Authentication on a Non-Windows Machine .....	28
Configuring TLS Verification on a Non-Windows Machine .....	29
Configuring Logging Options on a Non-Windows Machine .....	30
Testing the Connection on a Non-Windows Machine .....	33
Using a Connection String .....	35
DSN Connection String Example .....	35
DSN-less Connection String Examples .....	35
Features .....	38
Data Types .....	38
Security and Authentication .....	40
Driver Configuration Options .....	42
Configuration Options Appearing in the User Interface .....	42

## Installation and Configuration Guide

Configuration Options Having Only Key Names .....	49
Third-Party Trademarks .....	53
Third-Party Licenses .....	54

## About the Simba SQL Server ODBC Driver

The Simba SQL Server ODBC Driver enables Business Intelligence (BI), analytics, and reporting on data that is stored in Microsoft SQL Server 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 the *Data Access Standards Glossary*: <http://www.simba.com/resources/data-access-standards-library>. For complete information about the ODBC specification, see the *ODBC API Reference*: [http://msdn.microsoft.com/en-us/library/windows/desktop/ms714562\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms714562(v=vs.85).aspx).

The Simba SQL Server ODBC Driver is available for Microsoft® Windows®, Linux, and macOS platforms.

The *Installation and Configuration Guide* is suitable for users who are looking to access data residing within SQL Server 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](http://cdn.simba.com/docs/ODBC_QuickstartGuide/content/quick_start/intro.htm).

## Windows Driver

### Windows System Requirements

The Simba SQL Server ODBC Driver has been tested with SQL Server 2005 to 2014.

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

- One of the following operating systems:
  - Windows 7, 8.1, or 10
  - Windows Server 2008 or later
- 100 MB of available disk space
- Visual C++ Redistributable for Visual Studio 2013 installed (with the same bitness as the driver that you are installing).  
You can download the installation packages at <https://www.microsoft.com/en-ca/download/details.aspx?id=40784>.

To install the driver, you must have Administrator privileges on the machine.

### 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 the version of the driver that matches the bitness of the client application:

- `SimbaSQLServerODBC32.msi` for 32-bit applications
- `SimbaSQLServerODBC64.msi` for 64-bit applications

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

#### To install the Simba SQL Server ODBC Driver on Windows:

1. Depending on the bitness of your client application, double-click to run **SimbaSQLServerODBC32.msi** or **SimbaSQLServerODBC64.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**.
7. If you received a license file through email, then copy the license file into the `\lib` subfolder of the installation folder you selected above. You must have Administrator privileges when changing the contents of this folder.

## Creating a Data Source Name on Windows

Typically, after installing the Simba SQL Server 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 35.

### To create a Data Source Name on Windows:

1. Open the ODBC Administrator:
  - If you are using Windows 7 or earlier, click **Start**  > **All Programs** > **Simba SQL Server ODBC Driver 1.3** > **ODBC Administrator**.
  - Or, if you are using Windows 8 or later, on the Start screen, type **ODBC administrator**, and then click the **ODBC Administrator** search result.

 **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 SQL Server.


2. In the ODBC Data Source Administrator, click the **Drivers** tab, and then scroll down as needed to confirm that the Simba SQL Server 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 SQL Server ODBC Driver** and then click **Finish**. The Simba SQL Server 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 **Server** field, type the name or IP address of the host where your SQL Server instance is running.

 **Note:**

You can specify the name of the SQL Server instance by using the syntax **[ServerInfo][Instance]**, where **[ServerInfo]** is the IP address or host name of the server and **[Instance]** is the name of the instance.


9. 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 SQL Server is 1433.

10. In the **Database** field, type the name of the database that you want to access.
11. To use TLS to encrypt all communication with the SQL Server instance before sending it over the network, select the **Encrypt** check box and then do one of the following:
  - To enable one-way authentication so that the driver verifies the server certificate using a CA certificate, in the **CA Certificate** field, type the full path and file name of the CA certificate that you want to use.
  - Or, to trust the server certificate instead of verifying it, select the **Trust Server Certificate** check box.
12. Choose your authentication method:
  - To authenticate the connection using user login, type your credentials for accessing the server in the **User** and **Password** fields.
  - Or, to authenticate the connection using Kerberos, select the **Use Trusted Connection** check box. Optionally, in the **Server SPN** field, type the service principal name of your SQL Server instance. If you leave the field empty, then the driver uses **MSSQLSvc/[HostName]:[Port]** as the service principal name, where **[HostName]** is the IP address or host name of the server and **[Port]** is the number of the port that you are connecting to.
13. To return certain SQL Server-specific data types as ODBC data types instead of SQL Server data types, select the **Return SQL Server-Specific Types as ODBC Types** check box. For a complete list of the data types that this setting applies to, see [Return SQL Server-Specific Types as ODBC Types](#) on page 47.

14. To include temporary tables in the results when calling SQLTables, select the **Enable Table Types** check box.
15. To configure logging behavior for the driver, click **Logging Options**. For more information, see [Configuring Logging Options on Windows](#) on page 11.
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 SQL Server ODBC Driver DSN Setup dialog box are correct. Contact your SQL Server administrator as needed.

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

## Configuring Logging Options on Windows

To help troubleshoot issues, you can enable logging in the driver or in the wire protocol component. In addition to these forms of logging supported by the Simba SQL Server 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.

### Driver Logging

Use driver logging to track the activity in the Simba SQL Server ODBC Driver. You can specify the amount of detail included in the log files. The table below lists the logging levels that are available, in order from least verbose to most verbose.

Logging Level	Description
OFF	Disables all logging.
FATAL	Logs severe error events that lead the driver to abort.
ERROR	Logs error events that might allow the driver to continue running.

Logging Level	Description
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.


### 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. In the **Log Level** drop-down list, select the desired level of information to include in log files.
3. In the **Log Path** field, type the full path to the folder where you want to save log files.
4. If requested by Technical Support, type the name of the component for which to log messages in the **Log Namespace** field. Otherwise, do not type a value in the field.
5. 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.

6. 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.

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

The Simba SQL Server ODBC Driver produces a log file named `SqlServerODBC.log` at the location you specify using the Log Path field.

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 51.

### To disable 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. In the **Log Level** drop-down list, select **LOG\_OFF**.
3. Click **OK**.


## Wire Protocol Component Logging

Use wire protocol component logging to track the data activity between the driver and the SQL Server instance. You can specify the amount of detail to include in the log files. The table below lists the logging levels that are available, in order from least verbose to most verbose.

Logging Level	Description
0	Disables all logging.
1	Logs error events that might allow the wire protocol component to continue running. 1 is the default logging level.
2	Logs general information that describes the progress of the wire protocol component.
3	Logs detailed information that is useful for debugging the wire protocol component.
4	Logs all activity in the wire protocol component.

Wire protocol component logging is configured through the TDSTRACE environment variable.


### To enable wire protocol component logging on Windows:

1. Open the System Information dialog box:
  - If you are using Windows 7 or earlier, click **Start** , then right-click **Computer**, and then click **Properties**.

- Or, if you are using Windows 8 or later, on the Start screen, right-click **This PC** and then click **Properties**.
2. Click **Advanced System Settings**.
3. In the System Properties dialog box, click the **Advanced** tab and then click **Environment Variables**.
4. Choose one:
  - If the TDSTRACE variable already exists in the System Variables list, select it and then click **Edit**.
  - Or, if the TDSTRACE variable does not appear in the System Variables list, click **New** and then in the **Variable Name** field type **TDSTRACE**.
5. In the **Variable Value** field, type **[LoggingLevel]:[LogFilePath]**, where *[LoggingLevel]* is the logging level indicating the amount of detail to include in the log file and *[LogFilePath]* is the full path of the log file.

For example, the value **3:C:\Logs\MyWireLog.log** configures the wire protocol component to log debugging information in a file named `MyWireLog.log` located in the `C:\Logs` folder.
6. To save your changes and close the Edit System Variable dialog box, click **OK**.
7. Click **OK** to close the Environment Variables dialog box, and then click **OK** to close the System Properties dialog box.

#### To disable wire protocol component logging on Windows:

1. Open the System Information dialog box:
  - If you are using Windows 7 or earlier, click **Start** , then right-click **Computer**, and then click **Properties**.
  - Or, if you are using Windows 8 or later, on the Start screen, right-click **This PC** and then click **Properties**.
2. Click **Advanced System Settings**.
3. In the System Properties dialog box, click the **Advanced** tab and then click **Environment Variables**.
4. Select **TDSTRACE** from the System Variables list and then click **Edit**.
5. In the **Variable Value** field, replace the existing value with **0**.
6. To save your changes and close the Edit System Variable dialog box, click **OK**.
7. Click **OK** to close the Environment Variables dialog box, and then click **OK** to close the System Properties dialog box.

## ODBC Tracing

You can use the ODBC Data Source Administrator to trace connection activity in the ODBC layer.

**To start tracing using the ODBC Data Source Administrator:**

1. In the ODBC Data Source Administrator, click the **Tracing** tab.
2. In the Log File Path area, click **Browse**. In the Select ODBC Log File dialog box, browse to the location where you want to save the log file, then type a descriptive file name in the **File** name field, and then click **Save**.
3. On the Tracing tab, click **Start Tracing Now**.

**To stop ODBC Data Source Administrator tracing:**


- In the ODBC Data Source Administrator, on the Tracing tab, click **Stop Tracing Now**.


For more information about tracing using the ODBC Data Source Administrator, see "How to Generate an ODBC Trace with ODBC Data Source Administrator" on the Microsoft Support website: <http://support.microsoft.com/kb/274551>.

## Verifying the Driver Version Number on Windows

If you need to verify the version of the Simba SQL Server 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 Windows 7 or earlier, click **Start**  > **All Programs** > **Simba SQL Server ODBC Driver 1.3** > **ODBC Administrator**.
  - Or, if you are using Windows 8 or later, on the Start screen, type **ODBC administrator**, and then click the **ODBC Administrator** search result.

 **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 SQL Server.

2. Click the **Drivers** tab and then find the Simba SQL Server 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 SQL Server ODBC Driver has been tested with SQL Server 2005 to 2014.

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


- macOS version 10.9, 10.10, or 10.11
- 150 MB of available disk space
- iODBC 3.52.7 or later

### Installing the Driver on macOS

The Simba SQL Server ODBC Driver is available for macOS as a .dmg file named `SimbaSQLServerODBC.dmg`. The driver supports both 32- and 64-bit client applications.

#### To install the Simba SQL Server ODBC Driver on macOS:

1. Double-click **SimbaSQLServerODBC.dmg** to mount the disk image.
2. Double-click **SimbaSQLServerODBC.dmg** to mount the disk image.
3. In the installer, click **Continue**.
4. On the Software License Agreement screen, click **Continue**, and when the prompt appears, click **Agree** if you agree to the terms of the License Agreement.
5. Optionally, to change the installation location, click **Change Install Location**, then select the desired location, and then click **Continue**.

 **Note:**

By default, the driver files are installed in the `/Library/simba/sqlserver` directory.

6. To accept the installation location and begin the installation, click **Install**.
7. When the installation completes, click **Close**.
8. If you received a license file through email, then copy the license file into the `/lib` subfolder in the driver installation directory. You must have root privileges when changing the contents of this folder.

For example, if you installed the driver to the default location, you would copy the license file into the `/Library/simba/sqlserver/lib` folder.



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 20.

## Verifying the Driver Version Number on macOS

If you need to verify the version of the Simba SQL Server ODBC Driver that is installed on your macOS machine, you can query the version number through the Terminal.

### To verify the driver version number on macOS:

- At the Terminal, run the following command:

```
pkgutil --info com.simba.sqlserverodbc
```

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

## Linux Driver

### Linux System Requirements

The Simba SQL Server ODBC Driver has been tested with SQL Server 2005 to 2014.

Install the driver on client machines where the application is installed. Each 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 7 or 8
  - Ubuntu 14.04 or 16.04
- 150 MB of available disk space
- One of the following ODBC driver managers installed:
  - iODBC 3.52.7 or later
  - unixODBC 2.3.0 or later

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

### Installing the Driver Using the Tarball Package

The Simba SQL Server ODBC Driver is available as a tarball package named `SimbaSQL_ServerODBC-[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 the version of the driver that matches the bitness of the client application. You can install both versions of the driver on the same machine.

#### To install the Simba SQL Server ODBC 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 --directory=/opt -zxvf [TarballName]
```

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

The Simba SQL Server ODBC Driver files are installed in the `opt/simba/sqlserver` directory.

3. If you received a license file through email, then copy the license file into the `opt/simba/sqlserver/lib/32` or `opt/simba/sqlserver/lib/64` folder, depending on the version of the driver that you installed. You must have root privileges when changing the contents of this folder.

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 20.

## 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 SQL Server 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 20.
- 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 21.

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 23.

## 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.sqlserverodbc.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 `SIMBASQLSERVERINI` to the full path and file name of the `simba.sqlserverodbc.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 `SIMBASQLSERVERINI` to the full path and file name of the `simba.sqlserverodbc.ini` file.

For example, if your `odbc.ini` and `odbcinst.ini` files are located in `/usr/local/odbc` and your `simba.sqlserverodbc.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 SIMBASQLSERVERINI=/etc/simba.sqlserverodbc.ini
```

For unixODBC:

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

```
export SIMBASQLSERVERINI=/etc/simba.sqlserverodbc.ini
```

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

1. If the `SIMBASQLSERVERINI` 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.sqlserverodbc.ini`.
3. The driver searches the current working directory of the application for a file named `simba.sqlserverodbc.ini`.
4. The driver searches the home directory for a hidden file named `.simba.sqlserverodbc.ini` (prefixed with a period).
5. The driver searches the `/etc` directory for a file named `simba.sqlserverodbc.ini`.

## Configuring ODBC Connections on a Non-Windows Machine

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

- [Creating a Data Source Name on a Non-Windows Machine](#) on page 23
- [Configuring a DSN-less Connection on a Non-Windows Machine](#) on page 26
- [Configuring Authentication on a Non-Windows Machine](#) on page 28
- [Configuring TLS Verification on a Non-Windows Machine](#) on page 29
- [Configuring Logging Options on a Non-Windows Machine](#) on page 30
- [Testing the Connection on a Non-Windows Machine](#) on page 33


### 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 26.

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 SQL Server ODBC Driver
```

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

```
[ODBC Data Sources]
Sample DSN=Simba SQL Server 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/sqlserver/lib/libsqlserverodbc_
c_sbu.dylib
```

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

```
Driver=/opt/simba/sqlserver/lib/32/libsqlserverodbc_
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=1500
```

- c. Configure authentication by specifying the authentication mechanism and your credentials. For more information, see [Configuring Authentication on a Non-Windows Machine](#) on page 28.
      - d. Optionally, encrypt your connection with TLS and configure whether the driver verifies the identity of the server. For more information, see [Configuring TLS Verification on a Non-Windows Machine](#) on page 29.
      - e. 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 SQL Server ODBC Driver, see [Driver Configuration Options](#) on page 42.
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 21.

For example, the following is an `odbc.ini` configuration file for macOS containing a DSN that connects to SQL Server using a user account:

```
[ODBC Data Sources]
Sample DSN=Simba SQL Server ODBC Driver
[Sample DSN]
Driver=/Library/simba/sqlserver/lib/libsqlserverodbc_
sbu.dylib
Server=192.168.222.160
Port=1500
Trusted_Connection=no
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 SQL Server using a user account:

```
[ODBC Data Sources]
Sample DSN=Simba SQL Server ODBC Driver 32-bit
[Sample DSN]
Driver=/opt/simba/sqlserver/lib/32/libsqlserverodbc_sb32.so
Server=192.168.222.160
Port=1500
Trusted_Connection=no
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 SQL Server 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/sqlserver/lib/libsqlserverodbc_
c_sbu.dylib
```

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

```
Driver=/opt/simba/sqlserver/lib/32/libsqlserverodbc_
sb32.so
```

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

For example:

```
Description=Simba SQL Server 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 21.

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

```
[ODBC Drivers]
Simba SQL Server ODBC Driver=Installed
[Simba SQL Server ODBC Driver]
Description=Simba SQL Server ODBC Driver
Driver=/Library/simba/sqlserver/lib/libsqlserverodbc_
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 SQL Server ODBC Driver 32-bit=Installed
Simba SQL Server ODBC Driver 64-bit=Installed
[Simba SQL Server ODBC Driver 32-bit]
Description=Simba SQL Server ODBC Driver (32-bit)
Driver=/opt/simba/sqlserver/lib/32/libsqlserverodbc_sb32.so
[Simba SQL Server ODBC Driver 64-bit]
Description=Simba SQL Server ODBC Driver (64-bit)
Driver=/opt/simba/sqlserver/lib/64/libsqlserverodbc_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 35.

For instructions about configuring specific connection features, see the following:

- [Configuring Authentication on a Non-Windows Machine](#) on page 28
- [Configuring TLS Verification on a Non-Windows Machine](#) on page 29

For detailed information about all the connection properties that the driver supports, see [Driver Configuration Options](#) on page 42.

## Configuring Authentication on a Non-Windows Machine

Connections to SQL Server require authentication. When using the driver on a non-Windows machine, you can authenticate the connection using your SQL Server user account or the Kerberos protocol. Additionally, the macOS driver also supports authentication through the NTLM protocol.

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.

### Using a User Account

You can authenticate the connection by providing your user name and password for accessing the SQL Server instance.

#### To configure authentication using a user account:

1. Set the `Trusted_Connection` property to `no`.
2. Set the `UID` property to an appropriate user name for accessing the server.
3. Set the `PWD` property to the password corresponding to the user name that you provided above.

### Using Kerberos

You can authenticate the connection by using the Kerberos protocol.

#### To configure authentication using Kerberos:

1. Set the `Trusted_Connection` property to `yes`.
2. Set the `ServerSPN` property to the service principal name of the SQL Server instance.

 **Note:**

If you do not set the `ServerSPN` property, the driver uses `MSSQLSvc/[HostName]:[Port]` as the service principal name, where `[HostName]` is the IP address or host name of the server and `[Port]` is the number of the port that you are connecting to.

## Using NTLM

If you are using the driver on a macOS machine, you can authenticate the connection by using the NTLM protocol.

### To configure authentication using NTLM:

1. Set the `Trusted_Connection` property to `NTLM`.
2. Set the `UID` property to an appropriate user name for accessing the server.
3. Set the `PWD` property to the password corresponding to the user name that you provided above.

## Configuring TLS Verification on a Non-Windows Machine

If you are connecting to a SQL Server instance that has Transport Layer Security (TLS) enabled, then you can use TLS to encrypt your connection. When connecting to a server using TLS, the driver can be configured to verify the identity of the server. See [Security and Authentication](#) on page 40 for more details on TLS.

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 One-way TLS Verification

You can configure the driver to verify the identity of the server.

### To configure one-way TLS verification on a non-Windows machine:

1. To encrypt your connection using TLS, set the `Encrypt` property to `yes`.
2. To enable one-way authentication so that the driver verifies the identity of the server, set the `TrustServerCertificate` property to `no`.
3. To specify the CA certificate that you want to use to verify the server certificate, set the `CACertificate` property to the full path and file name of the CA certificate.

## Configuring a TLS Connection without Identity Verification

You can configure the driver to use TLS encryption without verifying the identity of the server.

### To configure a TLS connection without verification on a non-Windows machine:

1. To encrypt your connection using TLS, set the `Encrypt` property to `yes`.
2. To trust the server certificate instead of verifying it, set the `TrustServerCertificate` property to `yes`.

## Configuring Logging Options on a Non-Windows Machine

To help troubleshoot issues, you can enable logging in the driver or in the wire protocol component.

### ! 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.sqlserverodbc.ini` file, which apply to all connections that use the driver.

### Driver Logging

Use driver logging to track the activity in the Simba SQL Server ODBC Driver.


### To enable logging on a non-Windows machine:

1. Open the `simba.sqlserverodbc.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.

LogLevel Value	Description
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. For example:
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.sqlserverodbc.ini` configuration file.
8. Restart your ODBC application to make sure that the new settings take effect.

The Simba SQL Server ODBC Driver produces a log file named `SqlServerODBC.log` at the location you specify using the `LogPath` key.

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.

**To disable logging on a non-Windows machine:**

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

**Wire Protocol Component Logging**

Use wire protocol component logging to track the data activity between the driver and the SQL Server instance.

**To enable wire protocol component logging on a non-Windows machine:**

1. Open the `simba.sqlserverodbc.ini` configuration file in a text editor.
2. Set the `TDSTRACE` key to `[LoggingLevel]:[LogFilePath]`, where `[LoggingLevel]` is the logging level indicating the amount of detail to include in the log file and `[LogFilePath]` is the full path of the log file.

The following logging levels are supported:

Logging Level	Description
0	Disables all logging.
1	Logs error events that might allow the wire protocol component to continue running. 1 is the default logging level.
2	Logs general information that describes the progress of the wire protocol component.
3	Logs detailed information that is useful for debugging the wire protocol component.
4	Logs all activity in the wire protocol component.

For example, the following setting configures the wire protocol component to log debugging information in a file named `MyWireLog.log` located in the `C:\Logs` folder:

```
TDSTRACE=3:/localhome/employee/Documents/MyWireLog.log
```



3. Save the `simba.sqlserverodbc.ini` configuration file.
4. Restart your ODBC application to make sure that the new settings take effect.

#### To disable wire protocol component logging on a non-Windows machine:

1. Open the `simba.sqlserverodbc.ini` configuration file in a text editor.
2. Set the `TDSTRACE` key to 0.
3. Save the `simba.sqlserverodbc.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 35.

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 42.

### 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.

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

- *[PortNumber]* is the number of the TCP port that the SQL Server instance uses to listen for client connections.
- *[ServerInfo]* is the IP address or host name of the SQL Server instance to which you are connecting.
- *[ServiceName]* is the Kerberos service principal name of the SQL Server instance.
- *[SQLServerDatabase]* is the name of the database that you want to access.
- *[YourPassword]* is the password corresponding to your user name.

- *[YourUserName]* is the user name that you use to access the SQL Server instance.

## Connecting to a SQL Server Instance Using User Login

The following is an example of a DSN-less connection string for a connection to a SQL Server instance that requires user login:

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

For example:

```
Driver=Simba SQL Server ODBC Driver;  
Server=192.168.222.160;Port=1500;Database=TestDB;  
UID=simba;PWD=simba;
```

## Connecting to a SQL Server Instance Using Kerberos Authentication

The following is an example of a DSN-less connection string for a connection to a SQL Server instance that requires authentication through Kerberos:

```
Driver=Simba SQL Server ODBC Driver;Server=[ServerInfo];  
Port=[PortNumber];Database=[SQLServerDatabase];  
Integrated Security=SSPI;ServerSPN=[ServiceName];
```

For example:

```
Driver=Simba SQL Server ODBC Driver;  
Server=192.168.222.160;Port=1500;Database=TestDB;  
Integrated Security=SSPI;ServerSPN=sqls;
```

## Connecting to a SQL Server Instance Using NTLM Authentication

### Note:

NTLM authentication is available only in the macOS version of the driver.

The following is an example of a DSN-less connection string for a connection to a SQL Server instance that requires authentication through the NTLM protocol:

```
Driver=Simba SQL Server ODBC Driver;Server=[ServerInfo];  
Port=[PortNumber];Database=[SQLServerDatabase];
```

```
Integrated Security=NTLM;UID=[YourUserName];  
PWD=[YourPassword];
```

**For example:**

```
Driver=Simba SQL Server ODBC Driver;  
Server=192.168.222.160;Port=1500;Database=TestDB;  
Integrated Security=NTLM;UID=simba;PWD=simba;
```

## Features

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


- [Data Types](#) on page 38
- [Security and Authentication](#) on page 40


## Data Types

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

The table below lists the supported data type mappings.

SQL Server Type	SQL Type
BigInt	SQL_BIGINT
BigInt Identity	SQL_BIGINT
Binary	SQL_BINARY
Bit	SQL_BIT
Char	SQL_CHAR
Date	SQL_TYPE_DATE
DateTime	SQL_TYPE_TIMESTAMP
DateTime2	SQL_TYPE_TIMESTAMP
DateTimeOffset	SQL_SS_TIMESTAMPOFFSET
Decimal	SQL_DECIMAL
Decimal() Identity	SQL_DECIMAL
Float	SQL_FLOAT

SQL Server Type	SQL Type
GUID	SQL_GUID
Image	SQL_LONGVARBINARY
Int	SQL_INTEGER
Int Identity	SQL_INTEGER
Money	SQL_DECIMAL
NChar	SQL_WCHAR
NText	SQL_WLONGVARCHAR
Numeric	SQL_NUMERIC
Numeric() Identity	SQL_NUMERIC
NVarChar	SQL_WVARCHAR
NVarChar(Max)	SQL_WVARCHAR
Real	SQL_REAL
RowVersion	SQL_BINARY
 <b>Note:</b> Before SQL Server 2008, this data type was called Timestamp.	
SmallDateTime	SQL_TYPE_TIMESTAMP
SmallInt	SQL_SMALLINT
SmallInt Identity	SQL_INTEGER
SmallMoney	SQL_DECIMAL
SysName	SQL_WVARCHAR

SQL Server Type	SQL Type
Text	SQL_LONGVARCHAR
Time	SQL_SS_TIME2
Timestamp	SQL_BINARY
<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> <p> <b>Note:</b> In SQL Server 2008 and later, this data type is called RowVersion.</p> </div>	
TinyInt	SQL_TINYINT
TinyInt Identity	SQL_TINYINT
VarBinary	SQL_VARBINARY
VarBinary(Max)	SQL_VARBINARY
VarChar	SQL_VARCHAR
VarChar(Max)	SQL_VARCHAR

In addition, certain SQL Server data types are exposed via custom type identifiers, as in the native SQL Server driver:

- Geometry, Geography, and HierarchyID use custom type -151.
- XML uses custom type -152.

## Security and Authentication

To protect data from unauthorized access, SQL Server data stores require connections to be authenticated with user credentials and sometimes the TLS protocol. The Simba SQL Server ODBC Driver provides full support for these authentication protocols.

The driver provides mechanisms that enable you to authenticate your connection using the Kerberos protocol or your SQL Server user name and password. Additionally, the macOS driver supports authentication using the NTLM protocol. For detailed configuration instructions, see [Creating a Data Source Name on Windows](#) on page 9 or [Configuring Authentication on a Non-Windows Machine](#) on page 28.



Additionally, the driver supports TLS connections with or without one-way authentication using TLS. If the server has a TLS-enabled socket, then you can configure the driver to connect to it. The driver supports TLS 1.0 to 1.2. It will automatically detect the highest version the server supports and use that when negotiating the connection.

It is recommended that you enable TLS whenever you connect to a server that is configured to support it. TLS encryption protects data and credentials when they are transferred over the network, and provides stronger security than authentication alone. For detailed configuration instructions, see [Creating a Data Source Name on Windows](#) on page 9 or [Configuring TLS Verification on a Non-Windows Machine](#) on page 29.

## Driver Configuration Options

Driver Configuration Options lists the configuration options available in the Simba SQL Server 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 Simba SQL Server ODBC Driver DSN Setup dialog box.

When using a connection string or configuring a connection from a Linux or macOS 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 SQL Server ODBC Driver, or via the key name when using a connection string or configuring a connection from a Linux or macOS machine:

- [CA Certificate](#) on page 43
- [Database](#) on page 43
- [Enable Table Types](#) on page 43
- [Encrypt](#) on page 44
- [Log Level](#) on page 44
- [Log Path](#) on page 45
- [Max File Size](#) on page 45
- [Max Number Files](#) on page 46
- [Password](#) on page 46
- [Port](#) on page 47
- [Return SQL Server-Specific Types as ODBC Types](#) on page 47
- [Server](#) on page 47
- [Server SPN](#) on page 48
- [Trust Server Certificate](#) on page 48
- [Use Trusted Connection](#) on page 49
- [User](#) on page 49

## CA Certificate

Key Name	Default Value	Required
CACertificate	cacerts.pem	Yes, if the Encrypt option is enabled and the Trust Server Certificate option is disabled.

### Description

The full path and file name of the CA certificate that you want to use to verify the server certificate when TLS encryption is enabled.

For information about how to trust the server certificate instead of verifying it, see [Trust Server Certificate](#) on page 48.

## Database

Key Name	Default Value	Required
Database	None	No

### Description

The name of the SQL Server database that you want to access.

## Enable Table Types

Key Name	Default Value	Required
EnableTableTypes	Clear (no)	No

### Description

This option specifies whether the driver includes temporary tables in the results when calling SQLTables.

- Enabled (`yes`, `true`, or `1`): The driver includes temporary tables.
- Disabled (`no`, `false`, or `0`): The driver does not include temporary tables.

## Encrypt

Key Name	Default Value	Required
Encrypt	Clear (no)	No

### Description

This option specifies whether the driver uses TLS to encrypt communication with the SQL Server instance before sending it over the network. The driver supports TLS 1.0 to 1.2. It will automatically detect the highest version the server supports and use that when negotiating the connection.

- Enabled (`yes`, `true`, or `1`): The driver encrypts all communication with the SQL Server instance.
- Enabled (`yes`, `true`, or `1`): The driver does not encrypt communication with the SQL Server instance.

For information about configuring identity verification between the driver and the server, see [CA Certificate](#) on page 43 and [Trust Server Certificate](#) on page 48.

## 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.
- 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.sqlserverodbc.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 a log file named `SqlServerODBC.log` in the location specified in the Log Path (`LogPath`) property.

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 51.

## Log Path

Key Name	Default Value	Required
LogPath	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.sqlserverodbc.ini` file.

## 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.sqlserverodbc.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.sqlserverodbc.ini` file.

## Password

Key Name	Default Value	Required
PWD	None	Yes, if authenticating through user login or the NTLM protocol.

## Description

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

## Port

Key Name	Default Value	Required
Port	1433	Yes

## Description

The TCP port that the SQL Server instance uses to listen for client connections.

## Return SQL Server-Specific Types as ODBC Types

Key Name	Default Value	Required
ReturnSpecificType AsOdbcType	Clear (no)	No

## Description

This option specifies whether the driver returns the SQL Server-specific data types that are listed in the table below as ODBC data types, or as than as SQL Server data types.

- Enabled (*yes*, *true*, or *1*): The driver returns the data types listed below as ODBC data types.
- Disabled (*no*, *false*, or *0*): The driver returns the data types listed below as SQL Server data types.

SQL Server Type	Returned ODBC Type
TDS_SQL_SS_TIME	SQL_TYPE_TIME
TDS_SQL_SS_DATETIMEOFFSET	SQL_WVARCHAR

## Server

Key Name	Default Value	Required
Server	None	Yes

## Description

The host name or IP address of the SQL Server instance.

## Server SPN

Key Name	Default Value	Required
ServerSPN	MSSQLSvc/ [HostName]:[Port], where [HostName] is the server name or IP address specified in the Server option and [Port] is the port number specified in the Port option.	No

## Description

The service principal name of the SQL Server instance.

## Trust Server Certificate

Key Name	Default Value	Required
TrustServerCertificate	Clear (no)	No

## Description

This option specifies whether the driver trusts the server certificate when connecting to the server using TLS.

- Enabled (yes, true, or 1): The driver trusts the server certificate.
- Disabled (no, false, or 0): The driver does not trust the server certificate, and instead uses a CA certificate to verify the server certificate.

For information about how to specify a CA certificate, see [CA Certificate](#) on page 43.



## Use Trusted Connection

Key Name	Default Value	Required
Trusted_Connection OR Integrated Security	Clear (no or none)	No

### Description

This option specifies the protocol that the driver uses to authenticate connections to the SQL Server instance.

- **Enabled** (`Trusted_Connection` is set to `yes`, `true`, or `1`; or `Integrated Security` is set to `SSPI`): The driver uses the Kerberos protocol to authenticate connections.
- **Disabled** (`Trusted_Connection` is set to `no`, `false`, or `0`; or `Integrated Security` is not set): The driver uses user login to authenticate connections.

When using the macOS version of the driver, you can set the `Trusted_Connection` key or the `Integrated Security` key to `NTLM` to authenticate connections using the NTLM protocol.

### User

Key Name	Default Value	Required
UID	None	Yes, if authenticating through user login or the NTLM protocol.

### Description

The user name that you use to access the SQL Server instance.

## Configuration Options Having Only Key Names

The following configuration options do not appear in the Windows user interface for the Simba SQL Server ODBC Driver. They are accessible only when you use a connection string or configure a connection on macOS or Linux.

- [Driver](#) on page 50
- [LongColumnLength](#) on page 50
- [UseLogPrefix](#) on page 51

Additionally, the TDSTRACE option is available only as an environment variable.

- [TDSTRACE](#) on page 50

## Driver

Key Name	Default Value	Required
Driver	Simba SQL Server 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 SQL Server 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.

## LongColumnLength

Key Name	Default Value	Required
LongColumnLength	8000	No

## Description

The maximum size of BLOB and CLOB columns.

## TDSTRACE

Key Name	Default Value	Required
TDSTRACE	None	No

## Description

Use this property to enable or disable logging in the wire protocol component 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.
- This option is not supported in DSNs or connection strings. To configure wire protocol logging for the Windows driver, you must set TDSTRACE as an environment variable. To configure wire protocol logging for a non-Windows driver, you must set TDSTRACE in the `simba.sqlserverodbc.ini` file.

Set the property to `[LoggingLevel]:[LogFilePath]`, where `[LoggingLevel]` is the logging level indicating the amount of detail to include in the log file and `[LogFilePath]` is the full path of the log file.

The logging levels are supported:

- 0: Disable all logging.
- 1: Logs error events that might allow the wire protocol component to continue running.
- 2: Logs general information that describes the progress of the wire protocol component.
- 3: Logs detailed information that is useful for debugging the wire protocol component.
- 4: Logs all activity in the wire protocol component.

For example, setting TDSTRACE to `3:C:\Logs\MyWireLog.log` configures the wire protocol component to log debugging information in a file named `MyWireLog.log` located in the `C:\Logs` folder.

## 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.

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

## Third-Party Trademarks

Debian is a trademark or registered trademark of Software in the Public Interest, Inc. or its subsidiaries in Canada, United States and/or other countries.

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 SQL Server, SQL Server, 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.

Ubuntu is a trademark or registered trademark of Canonical Ltd. or its subsidiaries in Canada, United States and/or other countries.

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.

### **Libntlm License**

Libntlm is licensed under the [GNU Lesser General Public License](#) version 2.1 or (at your option) any later version.

### **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](mailto: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 adhered 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>