



SIMBA[®]
BY MAGNITUDE

Simba Netezza ODBC Driver with SQL Connector

Installation and Configuration Guide

Simba Technologies Inc.

Version 1.0.12
December 31, 2020



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

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

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

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

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

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

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

Contact Us

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

Tel: +1 (604) 633-0008

Fax: +1 (604) 633-0004

www.simba.com

About This Guide

Purpose

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

Audience

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

Knowledge Prerequisites

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

- Familiarity with the platform on which you are using the Simba Netezza ODBC Driver
- Ability to use the data source to which the Simba Netezza 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 Netezza 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 Advanced Options on Windows	11
Configuring SSL Verification on Windows	12
Configuring Driver Options on Windows	13
Configuring Driver-wide Logging Options on Windows	14
Setting Driver-Wide Configuration Options on Windows	16
Verifying the Driver Version Number on Windows	17
macOS Driver	18
macOS System Requirements	18
Installing the Driver Using the DMG File	18
Installing the Driver Using the Tarball Package	19
Verifying the Driver Version Number on macOS	19
Linux Driver	21
Linux System Requirements	21
Installing the Driver Using the Tarball Package	21
Installing the Driver Using the RPM File	22
Verifying the Driver Version Number on Linux	23
Configuring the ODBC Driver Manager on Non-Windows Machines	25
Specifying ODBC Driver Managers on Non-Windows Machines	25
Specifying the Locations of the Driver Configuration Files	26
Configuring ODBC Connections on a Non-Windows Machine	28
Creating a Data Source Name on a Non-Windows Machine	28
Configuring a DSN-less Connection on a Non-Windows Machine	31
Configuring SSL Verification on a Non-Windows machine	34
Configuring Logging Options on a Non-Windows Machine	35
Setting Driver-Wide Configuration Options	37
Testing the Connection on a Non-Windows Machine	37
Using a Connection String	40
DSN Connection String Example	40

Installation and Configuration Guide

DSN-less Connection String Examples	40
Features	42
Data Types	42
Security and Authentication	44
Driver Configuration Options	46
Configuration Options Appearing in the User Interface	46
Configuration Options Having Only Key Names	58
Contact Us	62
Third-Party Trademarks	63

About the Simba Netezza ODBC Driver

The Simba Netezza ODBC Driver enables Business Intelligence (BI), analytics, and reporting on data that is stored in Netezza 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 Netezza 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 Netezza ODBC Driver supports Netezza Performance Server 7.2.1.

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 or 8.1
 - Windows Server 2019, 2016, or 2012
 - 600 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

If you did not obtain this driver from the Simba website, you might need to follow a different installation procedure. For more information, see the *Simba OEM ODBC Drivers Installation Guide*.

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:

- `Simba Netezza 1.0 32-bit.msi` for 32-bit applications
- `Simba Netezza 1.0 64-bit.msi` for 64-bit applications

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

To install the Simba Netezza ODBC Driver on Windows:

1. Depending on the bitness of your client application, double-click to run **Simba Netezza 1.0 32-bit.msi** or **Simba Netezza 1.0 64-bit.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 Netezza ODBC Driver, you need to create a Data Source Name (DSN). A DSN is a data structure that stores connection information so that it can be used by the driver to connect to Netezza.

Alternatively, you can specify connection settings in a connection string or as driver-wide settings. Settings in the connection string take precedence over settings in the DSN, and settings in the DSN take precedence over driver-wide settings.

The following instructions describe how to create a DSN. For information about specifying settings in a connection string, see [Using a Connection String](#) on page 40. For information about driver-wide settings, see [Setting Driver-Wide Configuration Options on Windows](#) on page 16.

To create a Data Source Name on Windows:

1. From the Start menu, go to **ODBC Data Sources**.

 **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 Netezza.

2. In the ODBC Data Source Administrator, click the **Drivers** tab, and then scroll down as needed to confirm that the Simba Netezza 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 Netezza ODBC Driver** and then click **Finish**. The Simba Netezza 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 Netezza server.
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 Netezza is 5480.

10. In the **Database** field, type the service name of the Netezza database that you want to access.
11. In the **Schema** field, type the name of the Netezza schema to use.
12. Configure authentication by doing one of the following:
 - If your Netezza server is configured to authenticate the connection using Active Directory or MIT Kerberos, then in the **User Name** field, type your user name for accessing the database.
 - Or, if your Netezza server is configured to authenticate the connection using another authentication method, then in the **User Name** and **Password** fields, type your user name and password for accessing the database.

 **Note:**

Kerberos authentication on Netezza must be configured on the server.

13. To configure logging behavior for the driver, click **Logging Options**. For more information, see [Configuring Driver-wide Logging Options on Windows](#) on page 14.
14. To configure additional driver options, select a tab:

- For advanced driver options, see [Configuring Advanced Options on Windows](#) on page 11.
 - For SSL configuration, see [Configuring SSL Verification on Windows](#) on page 12.
 - For additional driver options, see [Configuring Driver Options on Windows](#) on page 13.
15. 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 Netezza ODBC Driver DSN Setup dialog box are correct. Contact your Netezza server administrator as needed.

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

Configuring Advanced Options on Windows

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

The following instructions describe how to configure advanced options in a DSN. You can specify the connection settings described below in a DSN, in a connection string, or as driver-wide settings. Settings in the connection string take precedence over settings in the DSN, and settings in the DSN take precedence over driver-wide settings.

To configure advanced options on Windows:

1. To access the advanced options, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, and then select the **Advanced DSN Options** tab.
2. To use the driver in read-only mode, select the **Read Only** check box.
3. To display the system tables used by the data store, select the **Show System Tables** check box.
4. To return SQL_BIT values as 1 or 0, select the **Return SQL_BIT as 1/0** check box.

Or, to return SQL_BIT values as t or f, clear the **Return SQL_BIT as 1/0** check box.

5. If you are using Kerberos authentication, then to use GSSAPI for authentication, select the **Use GSSAPI** check box.

6. Specify the date format by selecting one of the **Date Format** options:
 - **MDY**: The driver returns dates in MDY format, for example, 08-15-2019.
 - **DMY**: The driver returns dates in DMY format, for example, 15-08-2019.
 - **YMD**: The driver returns dates in YMD format, for example, 2019-08-15.
7. In the **Client User Id**, **Client Workstation Name**, **Client Application Name**, **Client Account String**, and **Client Program Info** fields, specify the client properties to send to the server when the session begins.
8. In the **Login Timeout** field, type the length of time, in seconds, before the login times out. To cause the login to never time out, type **0**.
9. In the **Query Timeout** field, type the length of time, in seconds, before the query times out. To cause the query to never time out, type **0**.
10. In the **Load Max Errors** field, type the maximum number of errors to accept during inserts with parameter arrays. Once the driver has received this many errors, the query fails.
11. To save your settings and close the Simba Netezza ODBC Driver DSN Setup dialog box, click **OK**.

Configuring SSL Verification on Windows

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

! Important:

The Simba Netezza ODBC Driver only supports SSL version 3. Other versions are not supported.

The following instructions describe how to configure SSL in a DSN. You can specify the connection settings described below in a DSN, in a connection string, or as driver-wide settings. Settings in the connection string take precedence over settings in the DSN, and settings in the DSN take precedence over driver-wide settings.

To configure SSL verification on Windows:

1. Configure SSL authentication on your Netezza database. For more information, see "Configuring the SSL Certificate" in the *IBM Knowledge Center*.
http://www.ibm.com/support/knowledgecenter/SSULQD_7.2.1/com.ibm.nz.adm.doc/t_sysadm_config_ssl_certs.html.
2. 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 DSN Options** tab.

3. For the Security Level, choose one:
 - To connect over an unsecured connection, select **Only Unsecured**. The driver does not connect to the data store if an unsecured connection is not available.
 - To connect over an unsecured connection if one is available, select **Preferred Unsecured**. The driver connects to the data store using an unsecured connection if available; if not, the driver uses a secure connection.
 - To connect over a secure connection if one is available, select **Preferred Secured**. The driver connects to the data store using a secure connection if available; if not, the driver uses an unsecured connection.
 - To connect over a secure connection, select **Only Secured**. The driver does not connect to the data store if a secure connection is not available.
 4. To specify the CA certificates that you want to use to verify the server, do one of the following:
 - To verify the server using the trusted CA certificates from a specific `.pem` file, specify the full path to the file in the **CA Certificate File** field and clear the **Use Windows Trust Store** check box.
 - Or, to use the trusted CA certificates `.pem` file that is installed with the driver, leave the **CA Certificate File** field empty, and clear the **Use Windows Trust Store** check box.
 - Or, to use the Windows Trust Store, select the **Use Windows Trust Store** check box.
- ! Important:**

 - If you are using the Windows Trust Store, make sure to import the trusted CA certificates into the Trust Store.
 - If you are using a specific CA certificate `.pem` file, make sure that the certificate is stored on the server.
5. To allow self-signed certificates from the server, select the **Allow Self-signed Certificates** check box.
 6. To allow expired certificates to authenticate the connection, select the **Allow Expired Certificates** check box.
 7. To allow the common name of a CA-issued SSL certificate to not match the host name of the Netezza server, select the **Allow Host Mismatch** check box.
 8. To save your settings and close the Simba Netezza ODBC Driver DSN Setup dialog box, click **OK**.

Configuring Driver Options on Windows

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

The following instructions describe how to configure driver options in a DSN. You can specify the connection settings described below in a DSN, in a connection string, or as driver-wide settings. Settings in the connection string take precedence over settings in the DSN, and settings in the DSN take precedence over driver-wide settings.

To configure driver options on Windows:

1. To access the driver options, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, and then select the **Driver Options** tab.
2. To specify the size of the communications buffer between the data store and the driver, in bytes, in the **Socket Buffer Size** field, type a number of bytes between 4096 and 131072.
3. To specify the number of rows to cache in memory at once, in the **Prefetch Count** field, type the number of rows.
4. To reset these values to their original defaults, click **Reset Defaults**.
5. To save your settings and close the Simba Netezza ODBC Driver DSN Setup dialog box, click **OK**.

Configuring Driver-wide Logging Options on Windows

To help troubleshoot issues, you can enable logging. In addition to functionality provided in the Simba Netezza 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 Netezza ODBC Driver, so make sure to disable the feature after you are done using it. To configure logging for the current connection, see [Setting Driver-Wide Configuration Options on Windows](#) on page 16.

To enable driver-wide 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.
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. You can type the path into the field, or click **Browse** and then browse to select the folder.
- 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 Netezza ODBC Driver produces the following log files at the location you specify in the Log Path field:

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

Setting Driver-Wide Configuration Options on Windows

When you specify connection settings in a DSN or connection string, those settings apply only when you connect to Netezza using that particular DSN or string. As an alternative, you can specify settings that apply to every connection that uses the Simba Netezza ODBC Driver by configuring them in the Windows Registry.

Note:

- Settings in the connection string take precedence over settings in the DSN, and settings in the DSN take precedence over driver-wide settings.
- If the `LogLevel` configuration property is passed in via the connection string or DSN, the rest of the logging configurations are read from the connection string or DSN and not from the existing driver-wide logging configuration.

To set driver-wide configuration options on Windows:

1. Choose one:
 - If you are using Windows 7 or earlier, click **Start** , then type **regedit** in the **Search** field, and then click **regedit.exe** in the search results.
 - Or, if you are using Windows 8 or later, on the Start screen, type **regedit**, and then click the **regedit** search result.

2. Navigate to the appropriate registry key for the bitness of your driver and your machine:

- If you are using the 32-bit driver on a 64-bit machine, then browse to the following registry key:

`HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Simba\Simba Netezza ODBC Driver\Driver`

- Otherwise, browse to the following registry key:

`HKEY_LOCAL_MACHINE\SOFTWARE\Simba\Simba Netezza ODBC Driver\Driver`

3. For each connection property that you want to configure, do the following:
 - a. Right-click the **Driver** subkey and then select **New > String Value**.
 - b. Type the key name of the connection property, and then press **Enter**.

For example, to specify the user name for authentication, type `UID`. To verify the supported key name for each driver configuration option, refer to the "Key Name" column in the description of the option in [Driver Configuration Options](#) on page 46.

- c. Right-click the value that you created in the previous steps and then click **Modify**.
- d. In the Edit String dialog box, in the **Value Data** field, type the value that you want to set the connection property to and then click **OK**.

For example, to specify "simba" as the user name for authentication, type `simba`.

4. Close the Registry Editor.

Verifying the Driver Version Number on Windows

If you need to verify the version of the Simba Netezza 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. From the Start menu, go to **ODBC Data Sources**.

 **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 Netezza.

2. Click the **Drivers** tab and then find the Simba Netezza 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 Netezza ODBC Driver supports Netezza Performance Server 7.2.1.

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 10.13 or 10.14
- 150MB of available disk space
- One of the following ODBC driver managers installed:
 - iODBC 3.52.9 or later
 - unixODBC 2.2.14 or later

Installing the Driver Using the DMG File

The Simba Netezza ODBC Driver is available for macOS as a `.dmg` file named `Simba Netezza 1.0.dmg`. The driver supports both 32- and 64-bit client applications.

To install the driver using the `.dmg` package:

1. Double-click **Simba Netezza 1.0.dmg** to mount the disk image.
2. In the installer, click **Continue**.
3. 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.
4. 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/netezzaodbc` directory.

5. To accept the installation location and begin the installation, click **Install**.
6. When the installation completes, click **Close**.
7. 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.

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 25.

Installing the Driver Using the Tarball Package

The Simba Netezza ODBC Driver is also available for macOS as a `.tar` file named `Simba Netezza 1.0.tar.gz`. The driver supports both 32- and 64-bit client applications.

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 --directory=/opt -zxvf [TarballName]
```

where, `[TarballName]` is the name of the tarball package containing the driver.

The Simba Netezza ODBC Driver files are installed in the `/opt/simba/netezzaodbc` directory.

3. If you received a license file through email, then copy the license file into the `/opt/simba/netezzaodbc/lib/32` or `/opt/simba/netezzaodbc/lib/64` folder, depending on the bitness of the driver that you have 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 25.

Verifying the Driver Version Number on macOS

If you need to verify the version of the Simba Netezza 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.netezzaodbc
```

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

Linux Driver

The Linux driver is available as an RPM file and as a tarball package.

Linux System Requirements

The Simba Netezza ODBC Driver supports Netezza Performance Server 7.2.1.

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) 7 or 8
 - CentOS 7 or 8
 - SUSE Linux Enterprise Server (SLES) 12 through 15
 - Debian 8 or 9
 - Ubuntu 14.04, 16.04, or 18.04
- 150 MB of available disk space
- One of the following ODBC driver managers installed:
 - iODBC 3.52.9 or later
 - unixODBC 2.2.14 or later

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

If you are using the RPM file to install the driver on Debian or Ubuntu, you must also have the `alien` utility installed. The `alien` utility is available on SourceForge: <https://sourceforge.net/projects/alien-pkg-convert/>.

Installing the Driver Using the Tarball Package

The Simba Netezza ODBC Driver is available as a tarball package named `SimbaNetezzaODBC-[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 --directory=/opt -zxvf [TarballName]
```

where, `[TarballName]` is the name of the tarball package containing the driver.

The Simba Netezza ODBC Driver files are installed in the `/opt/simba/netezzaodbc` directory.

3. If you received a license file through email, then copy the license file into the `opt/simba/netezzaodbc/lib/32` or `opt/simba/netezzaodbc/lib/64` folder, depending on the bitness of the driver that you have 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 25.

Installing the Driver Using the RPM File

If you did not obtain this driver from the Simba website, you might need to follow a different installation procedure. For more information, see the *Simba OEM ODBC Drivers Installation Guide*.

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:

- `simbanetezza-[Version]-[Release].i686.rpm` for the 32-bit driver
- `simbanetezza-[Version]-[Release].x86_64.rpm` for the 64-bit driver

The placeholders in the file names are defined as follows:

- `[Version]` is the version number of the driver.
- `[Release]` is the release number for this version of the driver.

You can install both the 32-bit and 64-bit versions of the driver on the same machine.

To install the Simba Netezza ODBC Driver using the RPM File:

1. Log in as the root user.
2. If you are installing the driver on a Debian or Ubuntu machine, download and install the `alien` utility:
 - a. Download the package from SourceForge:
<https://sourceforge.net/projects/alien-pkg-convert/>.
 - b. From the command line, run the following command:

```
sudo apt-get install alien
```

3. Navigate to the folder containing the RPM package for the driver.
4. Depending on the Linux distribution that you are using, run one of the following commands from the command line, where `[RPMFileName]` is the file name of the RPM package:

- If you are using Red Hat Enterprise Linux or CentOS, run the following command:

```
yum --nogpgcheck localinstall [RPMFileName]
```

- Or, if you are using SUSE Linux Enterprise Server, run the following command:

```
zypper install [RPMFileName]
```

- Or, if you are using Debian or Ubuntu, run the following command:

```
alien -i [RPMFileName]
```

The Simba Netezza ODBC Driver files are installed in the `/opt/simba/netezzaodbc` directory.

5. If you received a license file through email, then copy the license file into the `/opt/simba/netezzaodbc/lib/32` or `/opt/simba/netezzaodbc/lib/64` folder, depending on the version of the driver that you installed.

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 25.

Verifying the Driver Version Number on Linux

If you need to verify the version of the Simba Netezza 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 using the command-line interface:

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

- `yum list | grep SimbaNetezzaODBC`

- `rpm -qa | grep SimbaNetezzaODBC`

The command returns information about the Simba Netezza 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 Netezza 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 25.
- 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 26.

After configuring the ODBC driver manager, you can configure a connection and access your data store through the driver.

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.netezzaodbc.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_NETEZZA_ODBC_INI` to the full path and file name of the `simba.netezzaodbc.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_NETEZZA_ODBC_INI` to the full path and file name of the `simba.netezzaodbc.ini` file.

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

For unixODBC:

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

```
export SIMBA_NETEZZA_ODBC_INI=/etc/simba.netezzaodbc.ini
```

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

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

Configuring ODBC Connections on a Non-Windows Machine

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

- [Creating a Data Source Name on a Non-Windows Machine](#) on page 28
- [Configuring a DSN-less Connection on a Non-Windows Machine](#) on page 31
- [Configuring SSL Verification on a Non-Windows machine](#) on page 34
- [Configuring Logging Options on a Non-Windows Machine](#) on page 35
- [Setting Driver-Wide Configuration Options](#) on page 37
- [Testing the Connection on a Non-Windows Machine](#) on page 37

Creating a Data Source Name on a Non-Windows Machine

Typically, after installing the Simba Netezza ODBC Driver, you need to create a Data Source Name (DSN). A DSN is a data structure that stores connection information so that it can be used by the driver to connect to Netezza.

You can specify connection settings in a DSN (in the `odbc.ini` file), in a connection string, or as driver-wide settings (in the `simba.netezzaodbc.ini` file). Settings in the connection string take precedence over settings in the DSN, and settings in the DSN take precedence over driver-wide settings.

The following instructions describe how to create a DSN by specifying connection settings in the `odbc.ini` file. 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.

For information about specifying settings in a connection string, see [Configuring a DSN-less Connection on a Non-Windows Machine](#) on page 31 and [Using a Connection String](#) on page 40. For information about driver-wide settings, see [Setting Driver-Wide Configuration Options](#) on page 37.

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, then you need to remove the period (.) from the start of the file name before the file becomes editable.

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 Netezza ODBC Driver
```

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

```
[ODBC Data Sources]
Sample DSN=Simba Netezza 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/netezzaodbc/lib/libnetezzaodbc_sbu.dylib
```

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

```
Driver=/opt/simba/netezzaodbc/lib/32/libnetezzaodbc_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=5480
```

- c. Set the `Database` property to the name of the database that you want to access.

For example:

```
Database=Local
```

- d. Configure authentication by doing one of the following:
 - If your Netezza server is configured to authenticate the connection using MIT Kerberos or AD Kerberos, then set the `UID` property to your user name for accessing the database.

For example:

```
UID=skroob
```

- Or, if your Netezza server is configured to authenticate the connection using another authentication method, then set the `UID` and `PWD` properties to your user name and password for accessing the database.

For example:

```
UID=skroob  
PWD=simba123456
```

 **Note:**

Kerberos authentication on Netezza must be configured on the server.

- e. If you want to connect to the server through SSL, set the `SecurityLevel` connection property to the desired level of security, and set the `CACertFile` property to point to your security certificate. For more information, see [Configuring SSL Verification on a Non-Windows machine](#) on page 34.
 - f. 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 Netezza ODBC Driver, see [Driver Configuration Options](#) on page 46.
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 26.

For example, the following is an `odbc.ini` configuration file for macOS containing a DSN that connects to Netezza over Kerberos:

```
[ODBC Data Sources]
Sample DSN=Simba Netezza ODBC Driver
[Sample DSN]
Driver=/Library/simba/netezzaodbc/lib/libnetezzaodbc_
sbu.dylib
Server=192.168.222.160
Port=5480
Database=Local
UID=skroob
```

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 Netezza 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/netezzaodbc/lib/libnetezzaodbc_sbu.dylib
```

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

```
Driver=/opt/simba/netezzaodbc/lib/32/libnetezzaodbc_sb32.so
```

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

For example:

```
Description=Simba Netezza 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 ODBCYSINI environment variable specifies the location. For more information, see [Specifying the Locations of the Driver Configuration Files](#) on page 26.

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

```
[ODBC Drivers]
Simba Netezza ODBC Driver=Installed
[Simba Netezza ODBC Driver]
Description=Simba Netezza ODBC Driver
Driver=/Library/simba/netezzaodbc/lib/libnetezzaodbc_
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 Netezza ODBC Driver 32-bit=Installed
Simba Netezza ODBC Driver 64-bit=Installed
[Simba Netezza ODBC Driver 32-bit]
Description=Simba Netezza ODBC Driver (32-bit)
Driver=/opt/simba/netezzaodbc/lib/32/libnetezzaodbc_sb32.so
[Simba Netezza ODBC Driver 64-bit]
Description=Simba Netezza ODBC Driver (64-bit)
Driver=/opt/simba/netezzaodbc/lib/64/libnetezzaodbc_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 40.

For detailed information about all the connection properties that the driver supports, see [Driver Configuration Options](#) on page 46.

Configuring SSL Verification on a Non-Windows machine

If you are connecting to a Netezza server that has Secure Sockets Layer (SSL) enabled, you can configure the driver to connect to an SSL-enabled socket.

! Important:

The Simba Netezza ODBC Driver only supports SSL version 3. Other versions are not supported.

You can set the connection properties described below in a connection string, in a DSN (in the `odbc.ini` file), or as a driver-wide setting (in the `simba.netezzaodbc.ini` file). Settings in the connection string take precedence over settings in the DSN, and settings in the DSN take precedence over driver-wide settings.

To configure SSL verification on a non-Windows machine:

1. Configure SSL authentication on your Netezza database. For more information, see "Configuring the SSL Certificate" in the *IBM Knowledge Center*.
http://www.ibm.com/support/knowledgecenter/SSULQD_7.2.1/com.ibm.nz.adm.doc/t_sysadm_config_ssl_certs.html.
2. In your `odbc.ini` configuration file or connection string, set the `SecurityLevel` property to the level of SSL verification:
 - To connect over an unsecured connection, specify `onlyUnsecured`. The driver does not connect to the data store if an unsecured connection is not available.
 - To connect over an unsecured connection if one is available, specify `preferredUnsecured`. The driver connects to the data store using an unsecured connection if available; if not, the driver uses a secure connection.
 - To connect over a secure connection if one is available, specify `preferredSecured`. The driver connects to the data store using a secure connection if available; if not, the driver uses an unsecured connection.
 - To connect over a secure connection, specify `onlySecured`. The driver does not connect to the data store if a secure connection is not available.

For example, to connect only over a secure connection:

```
SecurityLevel=onlySecured
```

3. Set the `CaCertFile` property to the location of the CA `.pem` certificate file.

! Important:

The CA certificate must be stored on the server in the `/nz/` directory.

For example:

```
CaCertFile=/nz/CertFile.pem
```

4. To allow self-signed certificates from the server, set the `AllowSelfSignedCert` attribute to 1.
5. To allow the common name of a CA-issued SSL certificate to not match the host name of the Netezza server, set the `AllowHostMismatch` attribute to 1.
6. To allow expired certificates from the server, set the `AllowExpiredCert` attribute to 1.

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.

You can set the connection properties described below in a connection string, in a DSN (in the `odbc.ini` file), or as a driver-wide setting (in the `simba.netezzaodbc.ini` file). Settings in the connection string take precedence over settings in the DSN, and settings in the DSN take precedence over driver-wide settings.

To enable logging on a non-Windows machine:

1. 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.

2. Set the `LogPath` key to the full path to the folder where you want to save log files.
3. 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.

4. Set the `LogFileSize` key to the maximum size of each log file in bytes.

 **Note:**

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

5. Save the `simba.netezzaodbc.ini` configuration file.
6. Restart your ODBC application to make sure that the new settings take effect.

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

- A `simbanetezzaodbcdriver.log` file that logs driver activity that is not specific to a connection.
- A `simbanetezzaodbcdriver_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. Set the `LogLevel` key to 0.
2. Save the `simba.netezzaodbc.ini` configuration file.
3. Restart your ODBC application to make sure that the new settings take effect.

Setting Driver-Wide Configuration Options

When you specify connection settings in a DSN or connection string, those settings apply only when you connect to Netezza using that particular DSN or string. As an alternative, you can specify settings that apply to every connection that uses the Simba Netezza ODBC Driver by configuring them in the `simba.netezzaodbc.ini` file.



Note:

Settings in the connection string take precedence over settings in the DSN, and settings in the DSN take precedence over driver-wide settings.

To set driver-wide configuration options:

1. Navigate to the `lib` subfolder in the driver installation directory, and then open the `simba.netezzaodbc.ini` configuration file in a text editor.
2. In the `[Driver]` section, specify configuration options as key-value pairs. Start a new line for each key-value pair.

For example, to authenticate the connection using "simba" as the user name and "simba123" as the password, type the following:

```
UID=simba  
PWD=simba123
```

For detailed information about all the configuration options supported by the driver, see [Driver Configuration Options](#) on page 46.

3. Save the `simba.netezzaodbc.ini` configuration file.

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 40.

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 46.

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 examples are defined as follows:

- *[CertFile]* is the full path to the PEM certificate used by the server.
- *[DatabaseName]* is the name of the database that you want to access.
- *[PortNumber]* is the port that you use to access the server.
- *[SecLevel]* is the level of SSL/TLS security required by the driver. For example, PreferredSecured.
- *[ServerName]* is the name or IP address of the server 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 Netezza server.

Connecting to Netezza Using Your Netezza Database Credentials

The following is the format of a DSN-less connection string that connects to a Netezza server using your database credentials:

```
Driver=Simba Netezza ODBC Driver;Server=[ServerName];Port=[PortNumber];Database=[DatabaseName];UID=[YourUserName];PWD=[YourPassword]
```

For example:

```
Driver=Simba Netezza ODBC Driver;Server=192.168.222.160;Port=5480;Database=Local;UID=jsmith;PWD=simba123
```

Connecting to Netezza Using Kerberos

The following is the format of a DSN-less connection string for connecting to Netezza using the Kerberos protocol:

```
Driver=Simba Netezza ODBC Driver;Server=[ServerName];Port=[PortNumber];Database=[DatabaseName];UID=[YourUserName]
```

For example:

```
Driver=Simba Netezza ODBC Driver;Server=192.168.222.160;Port=5480;Database=Local;UID=jsmith
```

Connecting to Netezza Using SSL

The following is the format of a DSN-less connection string for connecting to Netezza and encrypting the connection using SSL. In this example, the driver authenticates the connection using Netezza database credentials; however, you can configure the driver to authenticate through Kerberos instead, as shown in the example above.

```
Driver=Simba Netezza ODBC Driver;Server=[ServerName];Port=[PortNumber];Database=[DatabaseName];UID=[YourUserName];PWD=[YourPassword];SecurityLevel=[SecLevel];CaCertFile=[CertFile]
```

For example:

```
Driver=Simba Netezza ODBC Driver;Server=192.168.222.160;Port=5480;Database=Local;UID=jsmith;PWD=simba123;SecurityLevel=PreferredSecured;CaCertFile=/nz/ca.pem
```

Features

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

- [Data Types](#) on page 42
- [Security and Authentication](#) on page 44

Data Types

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

! Important:

The maximum size for a record is 65,535 bytes.

The table below lists the supported data type mappings.

Netezza Type	Comment	SQLType
bigint (int8)	Signed	SQL_BIGINT
boolean (bool)	See Return SQL_BIT as 1/0 on page 55	SQL_BIT
byteint (int1)	Signed	SQL_TINYINT
char(n)	<ul style="list-style-type: none"> • Blank padded • Holds latin9 characters • n <= 64000 	SQL_CHAR
data slice	This is an internal data type.	SQL_SMALLINT
date	<ul style="list-style-type: none"> • Supports BCE dates • Year must be between 1 and 9999 inclusive 	SQL_TYPE_DATE

Netezza Type	Comment	SQLType
decimal(p,s)	Alias for numeric	SQL_NUMERIC
double precision (float8)		SQL_DOUBLE
integer (int4)	Signed	SQL_INTEGER
interval (timespan)	<ul style="list-style-type: none"> Internally stored as seconds Months are treated as 30 days 	SQL_INTERVAL_DAY_TO_SECOND
nchar(n)	<ul style="list-style-type: none"> Blank padded n <= 16000 	SQL_WCHAR
numeric(p, s)	<ul style="list-style-type: none"> p between 1 and 38 inclusive s between 0 and p inclusive 	SQL_NUMERIC
nvarchar(n)	n <= 16000	SQL_WVARCHAR
real (float4)		SQL_REAL
rowid	<ul style="list-style-type: none"> Internal type Returned as Bigint. 	SQL_BIGINT
smallint (int2)	Signed	SQL_SMALLINT
st_geometry(n)	n <= 64000	Not supported
time	<ul style="list-style-type: none"> Microsecond precision Does not support seconds > 59 	SQL_TYPE_TIME

Netezza Type	Comment	SQLType
time with timezone	<ul style="list-style-type: none"> • Microsecond precision • Time zone must be numeric • Offset does not support seconds > 59) 	SQL_VARCHAR
timestamp	<ul style="list-style-type: none"> • Microsecond precision • Does not support seconds > 59 	SQL_TYPE_TIMESTAMP
transaction id	<ul style="list-style-type: none"> • Internal type • Returned as Bigint. 	SQL_BIGINT
varbinary(n)	<ul style="list-style-type: none"> • n <= 64000 • Hex literals use x'FFFF' notation 	SQL_VARBINARY
varchar(n)	<ul style="list-style-type: none"> • Holds latin9 characters • n <= 64000 	SQL_VARCHAR

Security and Authentication

To protect data from unauthorized access, Netezza data stores may require connections to be authenticated with user credentials or the SSL protocol. The Simba Netezza ODBC Driver provides full support for all authentication protocols supported by your Netezza server. For information about configuring authentication on your Netezza server, see the *IBM Netezza System Administrator's Guide*.

If your Netezza server uses the MIT Kerberos or Active Directory Kerberos protocol, you only need to provide your Netezza user name. If your server uses a non-Kerberos authentication method such as LDAP, you must provide your Netezza user name and password. For information about how to specify your credentials for the connection, see [Creating a Data Source Name on Windows](#) on page 9 or [Creating a Data Source Name on a Non-Windows Machine](#) on page 28.

In addition, the driver supports SSL connections with one-way authentication using SSL version 3.

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 Verification on Windows](#) on page 12 or [Configuring SSL Verification on a Non-Windows machine](#) on page 34.

Driver Configuration Options

Driver Configuration Options lists the configuration options available in the Simba Netezza ODBC Driver alphabetically by field or button label.

When creating or configuring a connection on Windows, the fields and buttons are available in the Simba Netezza ODBC Driver DSN Setup dialog box.

When using a connection string, configuring driver-wide settings, or configuring a connection on macOS or Linux, use the key names provided.

Note:

Settings in the connection string take precedence over settings in the DSN, and settings in the DSN take precedence over driver-wide settings.

Configuration Options Appearing in the User Interface

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

- [Allow Common Name Host Name Mismatch](#) on page 47
- [Allow Expired Certificate](#) on page 47
- [Allow Self-Signed Server Certificate](#) on page 48
- [CA Certificate File](#) on page 48
- [Client Account String](#) on page 48
- [Client Application Name](#) on page 49
- [Client Program Information](#) on page 49
- [Client User ID](#) on page 49
- [Client Workstation Name](#) on page 49
- [Database](#) on page 50
- [Max File Size](#) on page 52
- [Max Number Files](#) on page 53
- [Password](#) on page 53
- [Port](#) on page 53
- [Prefetch Count](#) on page 54
- [Query Timeout](#) on page 54
- [Read Only](#) on page 54
- [Return SQL_BIT as 1/0](#) on page 55
- [Schema Name](#) on page 55
- [Security Level](#) on page 55
- [Server](#) on page 56
- [Show System Tables](#) on page 56
- [Socket Buffer Size](#) on page 56
- [Use GSSAPI](#) on page 57

- [Date Format](#) on page 50
- [Load Max Errors](#) on page 50
- [Log Level](#) on page 51
- [Log Path](#) on page 52
- [Login Timeout](#) on page 52
- [Use Windows Trust Store](#) on page 57
- [User Name](#) on page 57

Allow Common Name Host Name Mismatch

Key Name	Default Value	Required
AllowHostMismatch	Clear (0)	No

Description

This option specifies whether a CA-issued SSL certificate name must match the host name of the Netezza server.

- Enabled (1): The driver allows a CA-issued SSL certificate name to not match the host name of the Netezza server.
- Disabled (0): The CA-issued SSL certificate name must match the host name of the Netezza server.

Note:

This setting is applicable only when SSL is enabled.

Allow Expired Certificate

Key Name	Default Value	Required
AllowExpiredCert	Clear (0)	No

Description

This option specifies whether the driver allows expired certificates to be used to authenticate the connection.

- Enabled (1): The driver authenticates the Netezza server even if the server is using an expired certificate.
- Disabled (0): The driver does not allow expired certificates from the server.

Note:

This setting is applicable only when SSL is enabled.

Allow Self-Signed Server Certificate

Key Name	Default Value	Required
AllowSelfSigned Cert	Clear (0)	No

Description

This option specifies whether the driver allows a connection to a Netezza server that uses a self-signed certificate.

- Enabled (1): The driver authenticates the Netezza server even if the server is using a self-signed certificate.
- Disabled (0): The driver does not allow self-signed certificates from the server.

Note:

This setting is applicable only when SSL is enabled.

CA Certificate File

Key Name	Default Value	Required
CaCertFile	None	Yes, if SSL is enabled

Description

The full path to the SSL certificate that is used by the server.

Client Account String

Key Name	Default Value	Required
ClientAcctString	None	No

Description

The account string used by the client.

Client Application Name

Key Name	Default Value	Required
ClientApplName	None	No

Description

The application name used by the client.

Client Program Information

Key Name	Default Value	Required
ClientProgInfo	None	No

Description

The program information used by the client.

Client User ID

Key Name	Default Value	Required
ClientUserID	None	No

Description

The user ID used by the client.

Client Workstation Name

Key Name	Default Value	Required
ClientWorkStnName	None	No

Description

The workstation name used by the client.

Database

Key Name	Default Value	Required
Database	None	Yes

Description

The name of the Netezza database that you want to access.

Date Format

Key Name	Default Value	Required
DateFormat	YMD	No

Description

The preferred format for the driver to return dates.

- **YMD**: The driver returns dates in YMD format, for example, 2019-08-15.
- **DMY**: The driver returns dates in DMY format, for example, 15-08-2019.
- **MDY**: The driver returns dates in MDY format, for example, 08-15-2019.

Load Max Errors

Key Name	Default Value	Required
loadMaxErrors	1	No

Description

The maximum number of errors to accept during inserts with parameter arrays. Once the driver has received this many errors, the query fails.

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.
- When logging with connection strings and DSNs, this option only applies to per-connection logs.

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

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:

When logging with connection strings and DSNs, this option only applies to per-connection logs.

Login Timeout

Key Name	Default Value	Required
LoginTimeout	0	No

Description

The length of time, in seconds, before the login times out. A value of 0 (the default) indicates that the login never times out.

Max File Size

Key Name	Default Value	Required
LogFileSize	20971520	No

Description

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

If this property is set using the Windows UI, the entered value is converted from megabytes (MB) to bytes before being set.

! Important:

When logging with connection strings and DSNs, this option only applies to per-connection logs.

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:

When logging with connection strings and DSNs, this option only applies to per-connection logs.

Password

Key Name	Default Value	Required
PWD	None	Yes, if the Netezza server is not configured to use MIT Kerberos or Active Directory Kerberos for authentication.

Description

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

Port

Key Name	Default Value	Required
Port	5480	Yes

Description

The number of the TCP port that the Netezza uses to listen for client connections.

Prefetch Count

Key Name	Default Value	Required
PreFetch	5000	No

Description

The number of rows to cache in memory at once.

Query Timeout

Key Name	Default Value	Required
QueryTimeout	0	No

Description

The length of time, in seconds, before the query times out. A value of 0 (the default) indicates that the query never times out.

Read Only

Key Name	Default Value	Required
ReadOnly	Clear (0)	No

Description

This option controls whether the driver is in read-only mode.

- Enabled (1): The driver is in read-only mode, and cannot write to the data store.
- Disabled (0): The driver is not in read-only mode, and can write to the data store.

Return SQL_BIT as 1/0

Key Name	Default Value	Required
SQLBitOneZero	Clear (0)	No

Description

This option controls how SQL_BIT values are returned by the driver when a SQL_BIT is bound as a CHAR or WCHAR.

- Enabled (1): SQL_BIT values are returned as 1 or 0.
- Disabled (0): SQL_BIT values are returned as t or f.

Schema Name

Key Name	Default Value	Required
SchemaName	ADMIN	No

Description

The name of the schema that is used by the driver.

Security Level

Key Name	Default Value	Required
SecurityLevel	Preferred Unsecured (preferredUnSecured)	No

Description

The level of security (SSL/TLS) that the driver uses for the connection to the data store.

- Only Unsecured (onlyUnSecured): The driver does not use SSL.
- Preferred Unsecured (preferredUnSecured): If the server provides a choice, the driver does not use SSL.
- Preferred Secured (preferredSecured): If the server provides a choice, the driver uses SSL.

- Only Secured (`onlySecured`): The driver does not connect unless an SSL connection is available.

Server

Key Name	Default Value	Required
Server	None	Yes

Description

The host name or IP address of the Netezza server.

Show System Tables

Key Name	Default Value	Required
ShowSystemTables	Clear (0)	No

Description

This option controls whether the driver displays the system tables used by the data store.

- Enabled (1): The driver can display the data store system tables.
- Disabled (0): The driver does not display the system tables.

Socket Buffer Size

Key Name	Default Value	Required
Socket	8192	No

Description

The size of the socket communications buffer between the data store and the driver, in bytes. Specify a value from 4096 to 131072.

Use GSSAPI

Key Name	Default Value	Required
UseGSSAPI	Clear (0)	No

Description

This option indicates whether the driver should use GSSAPI with MIT Kerberos. To use this option, the MIT Kerberos library must be installed on the client machine. This option is only available on Windows, and is only used if the data source is using MIT Kerberos authentication.

- Enabled (1): The driver uses GSSAPI for Kerberos authentication.
- Disabled (0): The driver does not use GSSAPI for Kerberos authentication.

Use Windows 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 [CA Certificate File](#) on page 48.

Note:

This option is only available on Windows.

User Name

Key Name	Default Value	Required
UID	None	Yes

Description

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

Configuration Options Having Only Key Names

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

- [Driver](#) on page 58
- [DriverLocale](#) on page 59
- [Locale](#) on page 59
- [MaxCatalogNameLen](#) on page 59
- [MaxColumnNameLen](#) on page 60
- [MaxSchemaNameLen](#) on page 60
- [MaxTableNameLen](#) on page 60
- [UseLogPrefix](#) on page 61

Driver

Key Name	Default Value	Required
Driver	Simba Netezza 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 Netezza 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.

DriverLocale

Key Name	Default Value	Required
DriverLocale	en-US	No

Description

The locale to use for error messages.

This is a driver-wide setting, and cannot be specified in a DSN or connection string. To configure the locale on a per-session basis, see [Locale](#) on page 59.

If both `Locale` and `DriverLocale` are specified, `Locale` takes precedence.

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\SOFTWARE\Wow6432Node\Simba\Simba Netezza ODBC Driver\Driver`
- Otherwise: `HKEY_LOCAL_MACHINE\SOFTWARE\Simba\Simba Netezza ODBC Driver\Driver`

Use `DriverLocale` as the value name, and the locale as the value data.

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

Locale

Key Name	Default Value	Required
Locale	en-US	No

Description

The locale to use for error messages.

If both `Locale` and `DriverLocale` are specified, `Locale` takes precedence.

MaxCatalogNameLen

Key Name	Default Value	Required
MaxCatalogNameLen	128	No

Description

The maximum number of characters that can be returned for catalog names.

This option can be set to any integer from 0 to 65535, inclusive. To indicate that there is no maximum length or that the length is unknown, set this option to 0.

MaxColumnNameLen

Key Name	Default Value	Required
MaxColumnNameLen	128	No

Description

The maximum number of characters that can be returned for column names.

This option can be set to any integer from 0 to 65535, inclusive. To indicate that there is no maximum length or that the length is unknown, set this option to 0.

MaxSchemaNameLen

Key Name	Default Value	Required
MaxSchemaNameLen	128	No

Description

The maximum number of characters that can be returned for schema names.

This option can be set to any integer from 0 to 65535, inclusive. To indicate that there is no maximum length or that the length is unknown, set this option to 0.

MaxTableNameLen

Key Name	Default Value	Required
MaxTableNameLen	128	No

Description

The maximum number of characters that can be returned for table names.

This option can be set to any integer from 0 to 65535, inclusive. To indicate that there is no maximum length or that the length is unknown, set this option to 0.

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.

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 files would be named `jdoe_7836_simbanetezzaodbcdriver.log` and `jdoe_7836_simbanetezzaodbcdriver_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.

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\SOFTWARE\Wow6432Node\Simba\Simba Netezza ODBC Driver\Driver`
- Otherwise: `HKEY_LOCAL_MACHINE\SOFTWARE\Simba\Simba Netezza 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.netezzaodbc.ini` file.

Contact Us

If you have difficulty using the driver, please contact our Technical Support staff. We welcome your questions, comments, and feature requests.

 **Note:**

To help us assist you, prior to contacting Technical Support please prepare a detailed summary of the client and server environment including operating system version, patch level, and configuration.

You can contact Technical Support via the Magnitude Support Community at <http://magnitudesoftware.com/online-support/>.

You can also follow us on Twitter @SimbaTech and @Mag_SW

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.

All other trademarks are trademarks of their respective owners.