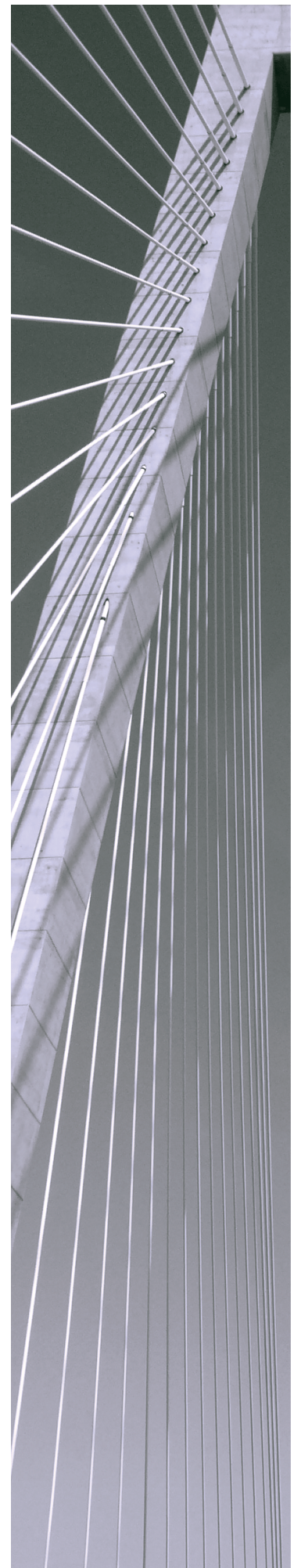# Simba PostgreSQL ODBC Driver

Installation and Configuration Guide

Simba Technologies Inc.

Version 1.3.7

September 27, 2017

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

Audience

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

Knowledge Prerequisites

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

- Familiarity with the platform on which you are using the Simba PostgreSQL ODBC Driver
- Ability to use the data source to which the Simba PostgreSQL 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 PostgreSQL ODBC Driver

The Simba PostgreSQL ODBC Driver enables Business Intelligence (BI), analytics, and reporting on data that is stored in PostgreSQL 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.

The Simba PostgreSQL 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 PostgreSQL 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

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
- 75 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:

- `Simba PostgreSQL 1.3 32-bit.msi` for 32-bit applications
- `Simba PostgreSQL 1.3 64-bit.msi` for 64-bit applications

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

To install the Simba PostgreSQL ODBC Driver on Windows:

1. Depending on the bitness of your client application, double-click to run **Simba PostgreSQL 1.3 32-bit.msi** or **Simba PostgreSQL 1.3 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**.

Creating a Data Source Name on Windows

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

Alternatively, for information about DSN-less connections, see Using a Connection String on page 37.

To create a Data Source Name on Windows:

1. Open the ODBC Administrator:
   - If you are using Windows 7 or earlier, click **Start** 🌐 **> All Programs > Simba PostgreSQL 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 PostgreSQL.

2. In the ODBC Data Source Administrator, click the **Drivers** tab, and then scroll down as needed to confirm that the Simba PostgreSQL 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 PostgreSQL ODBC Driver** and then click **Finish**. The Simba PostgreSQL ODBC Driver DSN Setup dialog box opens.

6. In the **Data Source** field, type a name for your DSN.

7. In the **Server** field, type the IP address or host name of the PostgreSQL server.

8. 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 PostgreSQL is 5432.

9. In the **Database** field, type the name of the database that you want to access.

10. In the **User** field, type your user name for accessing the PostgreSQL database.

11. In the **Password** field, type the password corresponding to the user name you typed.

12. Encrypt your credentials by selecting one of the following:

    - If the credentials are used only by the current Windows user, select **Current User Only**.
    - Or, if the credentials are used by all users on the current Windows machine, select **All Users Of This Machine**.

13. To configure client-server verification over SSL, click **SSL Options**. For more information, see Configuring SSL Verification on Windows on page 10.

14. To configure advanced driver options, click **Additional Options**. For more information, see Configuring Additional Options on Windows on page 12.

15. To configure logging behavior for the driver, click **Logging Options**. For more information, see Configuring Logging Options on Windows on page 15.

16. To configure how the driver returns and displays data, click **Data Type Options**. For more information, see Configuring Data Type Options on Windows on page 11.

17. 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 PostgreSQL ODBC Driver DSN Setup dialog box are correct. Contact your PostgreSQL server administrator as needed.

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

19. To close the ODBC Data Source Administrator, click **OK**.

Configuring SSL Verification on Windows

If you are connecting to a PostgreSQL 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.

To configure SSL verification on Windows:

1. To access the SSL options for a DSN, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, and then click **SSL Options**.

2. In the **Authentication Mode** list, select the appropriate SSL mode.

> ✎ **Note:**
>
> For information about SSL support in PostgreSQL, see "SSL Support" in the PostgreSQL Documentation: http://www.postgresql.org/docs/9.4/static/libpq-ssl.html.

3. To use the System Trust Store for SSL certificates, select the **Use System Trust Store** check box.

4. If you selected **Use System Trust Store**, choose on of the following options:
   - To check the validity of the certificate's trust chain, select the **Check Certificate Revocation** checkbox.
   - To accept self-signed certificates, select the**Allow Self-signed Server Certificate** checkbox.

5. To specify an SSL certificate, select the **Enable Custom SSL CA Root Certificate** check box, and then, in the **Path** field, specify the full path to the certificate file.

6. To save your settings and close the dialog box, click **OK**.

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

Configuring Data Type Options on Windows

You can configure data type options to modify how the driver displays or returns some data types.

To configure data type options on Windows:

1. To access data type options, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, and then click **Data Type Options**.

2. To enable the driver to return data as Unicode character types, select the **Use Unicode** check box.

> ✎ **Note:**
>
> When the **Use Unicode** check box is selected, the driver does the following:
> - Returns SQL_WCHAR instead of SQL_CHAR.
> - Returns SQL_WVARCHAR instead of SQL_VARCHAR.
> - Returns SQL_WLONGVARCHAR instead of SQL_LONGVARCHAR.

3. To configure the driver to return Boolean columns as SQL_VARCHAR instead of SQL_BIT, select the **Show Boolean Column As String** check box.

4. To configure the driver to return Text columns as SQL_LONGVARCHAR instead of SQL_VARCHAR, select the **Text as LongVarChar** check box.

5. To configure the driver to return Bytea columns as SQL_LONGVARBINARY instead of SQL_VARBINARY, select the **Bytea As LongVarBinary** check box.

6. In the **Max Varchar** field, type the maximum data length for VarChar columns.

7. In the **Max LongVarChar** field, type the maximum data length for LongVarChar columns.

8. In the **Max Bytea** field, type the maximum data length for Bytea columns.

9. To save your settings and close the Data Type Configuration dialog box, click **OK**.

Configuring Additional Options on Windows

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

To configure additional options on Windows:

1. To access advanced options, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, and then click **Additional Options**.

2. Specify how the driver processes queries by doing one of the following:
   - To return query results one row at a time, select **Single Row Mode**.
   - To return a specific number of rows at a time, select **Use Declare/Fetch** and then, in the **Cache Size** field, type the number of rows.
   - To enable the driver to have multiple queries active on the same connection, select **Use Multiple Statements**. The ODBC application may interleave calls to ODBC statements, but all queries are still sent and executed sequentially.
   - To return the entire query result, select **Retrieve Entire Result Into Memory**.

> ✎ **Note:**
>
> Use **Single Row Mode** if you plan to query large results and you do not want to retrieve the entire result into memory. Disabling **Single Row Mode** increases performance, but can result in out-of-memory errors.

3. To configure the driver to have only one active query at a time per connection, select **Enforce Single Statement Mode**.

4. To configure the driver to recognize table type information from the data source, select the **Enable Table Types** checkbox. For more information, see Enable Table Types on page 51.

5. If you are connecting through a proxy server, then select the **Enable HTTP Proxy Connection** check box and then do the following:

    a. In the **Proxy Server** field, type the host name or IP address of the proxy server.

    b. In the **Proxy Port** field, type the number of the TCP port that the proxy server uses to listen for client connections.

6. To save your settings and close the Advanced Options dialog box, click **OK**.

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

Configuring TCP Keepalives on Windows

By default, the Simba PostgreSQL ODBC Driver is configured to use TCP keepalives to prevent connections from timing out. Settings such as how frequently the driver sends TCP keepalive packets are based on the operating system defaults. You can configure the TCP keepalive settings or disable the feature by modifying the appropriate values in the Windows Registry.

To configure TCP keepalives 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. Select the appropriate registry key for the bitness of your driver:

    - If you are using the 32-bit driver on a 64-bit machine, then select the following registry key, where *[YourDSN]* is the DSN for which you want to configure keepalives:

        HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI*[YourDSN]*

- Otherwise, select the following registry key, where *[YourDSN]* is the DSN for which you want to configure keepalives:

  HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\\[YourDSN]

3. To specify the interval of inactivity before the driver sends a TCP keepalive packet, configure the **KeepAliveIdle** value by doing the following:

   a. If the **KeepAliveIdle** value does not already exist, create it. Select **Edit > New > String Value**, type **KeepAliveIdle** as the name of the value, and then press **Enter**.

   b. Select the **KeepAliveIdle** value, and then Select **Edit > Modify**.

   c. In the Edit String dialog box, in the **Value Data** field, type the number of seconds of inactivity before the driver sends a TCP keepalive packet.

   > ✎ **Note:**
   >
   > To use the system default, in the **Value Data** field, type **0**.

   d. Click **OK**.

4. To specify the number of TCP keepalive packets that can be lost before the connection is considered broken, configure the KeepAliveCount value. To do this, follow the procedure above, but type **KeepAliveCount** for the value name, and in the **Value Data** field, type the number of keepalive packets that can be lost.

   > ✎ **Note:**
   >
   > To use the system default, in the **Value Data** field, type **0**.

5. To specify the interval of time between each retransmission of a keepalive packet, configure the KeepAliveInterval value. To do this, follow the procedure above, but type **KeepAliveInterval** for the value name, and in the **Value Data** field, type the number of seconds to wait between each retransmission.

   > ✎ **Note:**
   >
   > To use the system default, in the **Value Data** field, type **0**.

6. Close the Registry Editor.

To disable TCP keepalives:

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. Select the appropriate registry key for the bitness of your driver:

   - If you are using the 32-bit driver on a 64-bit machine, then select the following registry key, where *[YourDSN]* is the DSN for which you want to configure keepalives:

     HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI*[YourDSN]*

   - Otherwise, select the following registry key, where *[YourDSN]* is the DSN for which you want to configure keepalives:

     HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI*[YourDSN]*

3. If the **KeepAlive** value does not already exist, create it. Select **Edit > New > String Value**, then type **KeepAlive** as the name of the value, and then press **Enter**.

4. Select the **KeepAlive** value, and then click **Edit > Modify**.

5. In the Edit String dialog box, in the **Value Data** field, type **0**.

6. Click **OK**.

7. Close the Registry Editor.

> ✎ **Note:**
>
> To enable TCP keepalives after disabling them, set KeepAlive to 1.

Configuring Logging Options on Windows

To help troubleshoot issues, you can enable logging. In addition to functionality provided in the Simba PostgreSQL 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 PostgreSQL ODBC Driver, so make sure to disable the feature after you are done using it.

To enable driver logging on Windows:

1. To access logging options, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, and then click **Logging Options**.

2. From the **Log Level** drop-down list, select the logging level corresponding to the amount of information that you want to include in log files:

Logging Level	Description
OFF	Disables all logging.
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.

3. In the **Log Path** field, specify the full path to the folder where you want to save log files.
4. Click **OK**.
5. Restart your ODBC application to make sure that the new settings take effect.

The Simba PostgreSQL ODBC Driver produces two log files at the location you specify in the Log Path field, where *[DriverName]* is the name of the driver:

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

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 62.

To disable driver logging on Windows:

1. Open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, and then click **Logging Options**.
2. From the **Log Level** drop-down list, select **LOG_OFF**.

3. Click **OK**.

4. Restart your ODBC application to make sure that the new settings take effect.

Verifying the Driver Version Number on Windows

If you need to verify the version of the Simba PostgreSQL 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 PostgreSQL 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 PostgreSQL.

2. Click the **Drivers** tab and then find the Simba PostgreSQL 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

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
- 215 MB of available disk space
- iODBC 3.52.7 or later

Installing the Driver on macOS

The Simba PostgreSQL ODBC Driver is available for macOS as a `.dmg` file named `Simba PostgreSQL 1.3.dmg`. The driver supports both 32- and 64-bit client applications.

To install the Simba PostgreSQL ODBC Driver on macOS:

1. Double-click **Simba PostgreSQL 1.3.dmg** to mount the disk image.
2. Double-click **Simba PostgreSQL 1.3.pkg** to run the installer.
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/postgresql` directory.

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

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 22.

Verifying the Driver Version Number on macOS

If you need to verify the version of the Simba PostgreSQL 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.postgresqlodbc
```

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

Linux Driver

Linux System Requirements

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 PostgreSQL ODBC Driver is available as a tarball package named `SimbaPostgreSQLODBC-[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 PostgreSQL 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 -zxvf [TarballName]
```

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

The Simba PostgreSQL ODBC Driver files are installed in the `opt/simba/postgresql` directory.

Next, configure the environment variables on your machine to make sure that the ODBC driver manager can work with the driver. For more information, see Configuring the ODBC Driver Manager on Non-Windows Machines on page 22.

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 PostgreSQL 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 22.
- 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 23.

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 25.

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.postgresqlodbc.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 SIMBAPOSTGRESQLODBCINI to the full path and file name of the
  `simba.postgresqlodbc.ini` file.

> ✎ **Note:**
>
> If you accquired the driver from a vendor other than Simba, you need to
> replace SIMBA with the name of your vendor.

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 SIMBAPOSTGRESQLODBCINI to the full path and file name of the
  `simba.postgresqlodbc.ini` file.

> ✎ **Note:**
>
> If you accquired the driver from a vendor other than Simba, you need to
> replace SIMBA with the name of your vendor.

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

For unixODBC:

```
export ODBCINI=/usr/local/odbc/odbc.ini
export ODBCSYSINI=/usr/local/odbc
export SIMBAPOSTGRESQLODBCINI=/etc/simba.postgresqlodbc.ini
```

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

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

Configuring ODBC Connections on a Non-Windows Machine

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

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 28.

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 PostgreSQL ODBC Driver
```

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

```
[ODBC Data Sources]
Sample DSN=Simba PostgreSQL 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/postgresql/lib/libpostgresqlod
   bc_sbu.dylib
   ```

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

   ```
   Driver=/opt/simba/postgresql/lib/32/libpostgresqlodb
   c_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=5432
   ```

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

   For example:

   ```
   Database=TestDB
   ```

   d. To configure authentication, set the `UID` property to an appropriate user name for accessing the PostgreSQL server, and set the `PWD` property to the password corresponding to the user name you provided.

   For example:

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

e. To connect to the server through SSL, enable SSL and specify the certificate information. For more information, see Configuring SSL Verification on a Non-Windows Machine on page 30.

f. Optionally, modify how the driver runs queries and retrieves results into memory. For more information, see Configuring Query Processing Modes on a Non-Windows Machine on page 31.

g. Optionally, modify the TCP keepalive settings that the driver uses to prevent connections from timing out. For more information, see Configuring TCP Keepalives on a Non-Windows Machine on page 32.

h. 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 PostgreSQL ODBC Driver, see Driver Configuration Options on page 47.

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 23.

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

```
[ODBC Data Sources]
Sample DSN=Simba PostgreSQL ODBC Driver
[Sample DSN]
Driver=/Library/simba/postgresql/lib/libpostgresqlodbc_
sbu.dylib
Host=192.168.222.160
Port=5432
Database=TestDB
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 PostgreSQL:

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

```
Driver=/opt/simba/postgresql/lib/32/libpostgresqlodbc_
sb32.so
Host=192.168.222.160
Port=5432
Database=TestDB
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 PostgreSQL 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/postgresql/lib/libpostgresqlod
bc_sbu.dylib
```

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

```
Driver=/opt/simba/postgresql/lib/32/libpostgresqlodb
c_sb32.so
```

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

For example:

```
Description=Simba PostgreSQL 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 23.

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

```
[ODBC Drivers]
Simba PostgreSQL ODBC Driver=Installed
[Simba PostgreSQL ODBC Driver]
Description=Simba PostgreSQL ODBC Driver
Driver=/Library/simba/postgresql/lib/libpostgresqlodbc_
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 PostgreSQL ODBC Driver 32-bit=Installed
Simba PostgreSQL ODBC Driver 64-bit=Installed
[Simba PostgreSQL ODBC Driver 32-bit]
Description=Simba PostgreSQL ODBC Driver (32-bit)
Driver=/opt/simba/postgresql/lib/32/libpostgresqlodbc_
sb32.so
```

```
[Simba PostgreSQL ODBC Driver 64-bit]
Description=Simba PostgreSQL ODBC Driver (64-bit)
Driver=/opt/simba/postgresql/lib/64/libpostgresqlodbc_
sb64.so
```

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

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

- Configuring SSL Verification on a Non-Windows Machine on page 30
- Configuring Query Processing Modes on a Non-Windows Machine on page 31
- Configuring TCP Keepalives on a Non-Windows Machine on page 32

For detailed information about all the connection properties that the driver supports, see Driver Configuration Options on page 47.

Configuring SSL Verification on a Non-Windows Machine

If you are connecting to a PostgreSQL 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.

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.

To configure SSL verification on a non-Windows machine:

1. Set the `SSLMode` property to the appropriate SSL mode.

   > 📝 **Note:**
   >
   > For information about SSL support in PostgreSQL, see "SSL Support" in the PostgreSQL Documentation: http://www.postgresql.org/docs/9.4/static/libpq-ssl.html.

2. To use the System Trust Store for SSL certificates, set `UseSystemTrustStore` to `1`.

3. If you set `UseSystemTrustStore` above, choose on of the following options:
    - To check the validity of the certificate's trust chain, the `CheckCertRevocation` to `1`.
    - To accept self-signed certificates, check the `CheckCertRevocation` to `1`.
4. To specify an SSL certificate, set the `SSLCertPath` property to the full path and file name of the certificate file.

Configuring Query Processing Modes on a Non-Windows Machine

To optimize driver performance, you can modify how the driver runs queries and retrieves results into memory. For example, you can configure the driver to return entire query results into memory all at once, or one row at a time. Use a query processing mode that prevents queries from consuming too much memory, based on the expected result size of your queries and the specifications of your system.

> ✎ **Note:**
>
> Use Single Row Mode if you plan to query large results and you do not want to retrieve the entire result into memory. Using the other query processing modes increases performance, but can result in out-of-memory errors.

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.

Enabling Single Row Mode

You can configure the driver to return query results one row at a time.

To enable Single Row Mode:

1. Set the `SingleRowMode` property to `1`.
2. Make sure that the `UseDeclareFetch` property is set to `0` or not set.

Enabling Declare/Fetch Mode

You can configure the driver to return a specific number of rows at a time.

To enable Declare/Fetch Mode:

1. Set the `UseDeclareFetch` property to `1`.
2. Set the `Fetch` property to the number of rows that the driver returns at a time.

Enabling Retrieve Entire Result Mode

You can configure the driver to return entire query results into memory.

To enable Retrieve Entire Result Mode:

➤ Make sure that the `SingleRowMode`, `UseDeclareFetch`, and `UseMultipleStatements` properties are set to `0` or not set.

Enabling Multiple Statements Mode

You can enable the driver to have multiple queries active on the same connection. The ODBC application may interleave calls to ODBC statements, but all queries are still sent and executed sequentially. When using this mode, the driver returns all the query results into memory.

To enable Multiple Statements Mode:

1. Set the `UseMultipleStatements` property to `1`.
2. Make sure that the `SingleRowMode` and `UseDeclareFetch` properties are set to `0` or not set.

Configuring TCP Keepalives on a Non-Windows Machine

By default, the Simba PostgreSQL ODBC Driver is configured to use TCP keepalives to prevent connections from timing out. Settings such as how frequently the driver sends TCP keepalive packets are based on the operating system defaults.

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.

To configure TCP keepalives on a non-Windows machine:

1. Set the `KeepAliveIdle` property to the number of seconds of inactivity before the driver sends a TCP keepalive packet.
2. Set the `KeepAliveCount` property to the number of keepalive packets that can be lost before the connection is considered broken.
3. Set the `KeepAliveInterval` property to the number of seconds to wait before each retransmission of a keepalive packet.

> ✎ **Note:**
>
> To use the system default for `KeepAliveIdle`, `KeepAliveCount`, or `KeepAliveInterval`, set the property to `0`.

To disable TCP keepalives:

➢ Set the `KeepAlive` property to `0`.

> ✎ **Note:**
>
> To enable TCP keepalives after disabling them, remove the `KeepAlive` property or set it 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.

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

To enable logging on a non-Windows machine:

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

LogLevel Value	Description
0	Disables all logging.
1	Logs severe error events that lead the driver to abort.
2	Logs error events that might allow the driver to continue running.

LogLevel Value	Description
3	Logs events that might result in an error if action is not taken.
4	Logs general information that describes the progress of the driver.
5	Logs detailed information that is useful for debugging the driver.
6	Logs all driver activity.

3. Set the `LogPath` key to the full path to the folder where you want to save log files.
4. Optionally, to prefix the log file name with the user name and process ID associated with the connection, set the `UseLogPrefix` property to `1`.
5. Save the `simba.postgresqlodbc.ini` configuration file.
6. Restart your ODBC application to make sure that the new settings take effect.

The Simba PostgreSQL ODBC Driver produces two log files at the location you specify using the `LogPath` key, where *[DriverName]* is the name of the driver:

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

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

Testing the Connection on a Non-Windows Machine

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

Using the iODBC Driver Manager

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

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

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

To test your connection using the iODBC driver manager:

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

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

Using the unixODBC Driver Manager

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

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

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

To test your connection using the unixODBC driver manager:

➢ Run isql or iusql by using the corresponding syntax:

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

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

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

> ✎ **Note:**
>
> For information about the available options, run isql or iusql without providing a DSN.

Using a Connection String

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

The connection strings in the following sections are examples showing the minimum set of connection attributes that you must specify to successfully connect to the data source. Depending on the configuration of the data source and the type of connection you are working with, you might need to specify additional connection attributes. For detailed information about all the attributes that you can use in the connection string, see Driver Configuration Options on page 47.

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, in alphabetical order:

- *[PortNumber]* is the number of the TCP port that the PostgreSQL server uses to listen for client connections.
- *[PPort]* is the number of the TCP port that the proxy server uses to listen for client connections.
- *[PServer]* is the IP address or host name of the proxy server to which you are connecting.
- *[DatabaseName]* is the database that you want to access.
- *[Server]* is the IP address or host name of the PostgreSQL server to which you are connecting.

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

Connecting to a PostgreSQL Server Directly

The following is the format of a DSN-less connection string for a basic connection to a PostgreSQL server:

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

For example:

```
Driver=Simba PostgreSQL ODBC Driver;Server=192.168.222.160;
Port=5432;Database=TestDB;UID=simba;PWD=simba;
```

Connecting to a PostgreSQL Server through a Proxy Server

The following is the format of a DSN-less connection string for connecting to a PostgreSQL server through a proxy server:

```
Driver=Simba PostgreSQL ODBC Driver;
Server=[Server];Port=[PortNumber];
Database=[DatabaseName];UID=[YourUserName];
PWD=[YourPassword];ProxyHost=[PServer];ProxyPort=[PPort];
```

For example:

```
Driver=Simba PostgreSQL ODBC Driver;Server=192.168.222.160;
Port=5432;Database=TestDB;UID=simba;PWD=simba;
ProxyHost=192.168.222.160;ProxyPort=8000;
```

Features

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

- Query Processing Modes on page 39
- TCP Keepalives on page 40
- Data Types on page 40
- Security and Authentication on page 45

Query Processing Modes

To support performance tuning, the Simba PostgreSQL ODBC Driver provides different query processing modes that you can configure to modify how the driver runs queries and retrieves results into memory. The following query processing modes are available:

- **Single Row Mode**: The driver returns query results one row at a time.
- **Declare/Fetch Mode**: The driver returns a user-specified number of rows at a time.
- **Retrieve Entire Result Mode**: The driver returns the entire query result into memory.
- **Enforce Single Statement Mode:** The driver allows only one active statement at a time for each connection.
- **Multiple Statements Mode**: The driver can have multiple queries active on the same connection. The ODBC application may interleave calls to ODBC statements, but all queries are still sent and executed sequentially. When using this mode, the driver returns all the query results into memory.

By default, the driver returns the entire query result into memory.

Use a query processing mode that prevents queries from consuming too much memory, considering the expected result size of your queries and the specifications of your system.

For information about configuring how the driver processes queries, see Configuring Additional Options on Windows on page 12 if you are using the Windows version of the driver, or see Configuring Query Processing Modes on a Non-Windows Machine on page 31 if you are using a non-Windows version of the driver.

TCP Keepalives

By default, the Simba PostgreSQL ODBC Driver is configured to use TCP keepalives to verify the status of a connection and prevent it from timing out. After you connect to a PostgreSQL server, the driver automatically sends keepalive packets to the server. If the server does not respond, then the driver returns an indication that the connection is broken.

For information about configuring settings for TCP keepalives when using the Windows driver, see Configuring TCP Keepalives on Windows on page 13. For information about configuring settings for TCP keepalives when using the Linux or macOS driver, see Configuring TCP Keepalives on a Non-Windows Machine on page 32.

Data Types

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

The table below lists the supported data type mappings.

> ✎ **Note:**
>
> If the Use Unicode option (the `UseUnicode` key) is enabled, then the driver returns SQL_WCHAR instead of SQL_CHAR, and SQL_WVARCHAR instead of SQL_VARCHAR.

PostgreSQL Type	SQL Type
Array	SQL_VARCHAR
BigInt	SQL_BIGINT
BigSerial	SQL_BIGINT

PostgreSQL Type	SQL Type
Bit	SQL_VARCHAR • If the length of the column is greater than the Max Varchar (`MaxVarchar`) setting, then SQL_LONGVARCHAR is returned instead. • If the Use Unicode option (the `UseUnicode` key) is enabled, then SQL_WVARCHAR is returned instead. • If the Use Unicode option (the `UseUnicode` key) is enabled and the column length is greater than the Max Varchar (`MaxVarchar`) setting, then SQL_WLONGVARCHAR is returned instead.
Bit Varying	SQL_VARCHAR • If the length of the column is greater than the Max Varchar (`MaxVarchar`) setting, then SQL_LONGVARCHAR is returned instead. • If the Use Unicode option (the `UseUnicode` key) is enabled, then SQL_WVARCHAR is returned instead. • If the Use Unicode option (the `UseUnicode` key) is enabled and the column length is greater than the Max Varchar (`MaxVarchar`) setting, then SQL_WLONGVARCHAR is returned instead.
Boolean	SQL_BIT If the Show Boolean Column As String option (the `BoolsAsChar` key) is enabled, then SQL_VARCHAR is returned instead.
Box	SQL_VARCHAR
Bytea (escape and hex formats)	SQL_VARBINARY If the Bytea As LongVarBinary option (the `ByteaAsLongVarBinary` key) is enabled, then SQL_LONGVARBINARY is returned instead.

PostgreSQL Type	SQL Type
Char	SQL_CHAR • If the length of the column is greater than the Max Varchar (`MaxVarchar`) setting, then SQL_LONGVARCHAR is returned instead. • If the Use Unicode option (the `UseUnicode` key) is enabled, then SQL_WCHAR is returned instead. • If the Use Unicode option (the `UseUnicode` key) is enabled and the column length is greater than the Max Varchar (`MaxVarchar`) setting, then SQL_WLONGVARCHAR is returned instead.
Cid	SQL_VARCHAR
Cidr	SQL_VARCHAR
Circle	SQL_VARCHAR
Composite Types	SQL_VARCHAR
Date	SQL_TYPE_DATE
DateRange	SQL_VARCHAR
Decimal	SQL_NUMERIC
Double Precision	SQL_DOUBLE
Enum	SQL_VARCHAR
Float (same as Double Precision)	SQL_DOUBLE
Int4Range	SQL_VARCHAR
Int8Range	SQL_VARCHAR
Inet	SQL_VARCHAR
Integer	SQL_INTEGER

PostgreSQL Type	SQL Type
Interval	SQL_VARCHAR
Json	SQL_VARCHAR
JsonB	SQL_VARCHAR
Line	SQL_VARCHAR
LSeg	SQL_VARCHAR
MacAddr	SQL_VARCHAR
Money	SQL_FLOAT
Numeric	SQL_NUMERIC
NumRange	SQL_VARCHAR
Oid	SQL_INTEGER
Path	SQL_VARCHAR
Pg_Lsn	SQL_VARCHAR
Point	SQL_VARCHAR
Polygon	SQL_VARCHAR
Real	SQL_REAL
Serial	SQL_INTEGER
SmallInt	SQL_SMALLINT
SmallSerial	SQL_SMALLINT

PostgreSQL Type	SQL Type
Text	SQL_VARCHAR • If the Use Unicode option (the `UseUnicode` key) is enabled, then SQL_WVARCHAR is returned instead. • If the Text As LongVarChar option (the `TextAsLongVarchar` key) is enabled, then SQL_LONGVARCHAR is returned instead. • If both options are enabled, then SQL_WLONGVARCHAR is returned instead.
Tid	SQL_VARCHAR
Time (with time zone)	SQL_VARCHAR
Time (without time zone)	SQL_TYPE_TIME
Timestamp	SQL_TYPE_TIMESTAMP or SQL_TIMESTAMP (ODBC 2.0)
TsQuery	SQL_VARCHAR
TsRange	SQL_VARCHAR
TstzRange	SQL_VARCHAR
TsVector	SQL_VARCHAR
Txid_Snapshot	SQL_VARCHAR
Uuid	SQL_GUID

PostgreSQL Type	SQL Type
VarChar	SQL_VARCHAR • If the length of the column is greater than the Max Varchar (`MaxVarchar`) setting, then SQL_LONGVARCHAR is returned instead. • If the Use Unicode option (the `UseUnicode` key) is enabled, then SQL_WVARCHAR is returned instead. • If the Use Unicode option (the `UseUnicode` key) is enabled and the column length is greater than the Max Varchar (`MaxVarchar`) setting, then SQL_WLONGVARCHAR is returned instead.
Xid	SQL_INTEGER

Security and Authentication

To protect data from unauthorized access, PostgreSQL data stores require all connections to be authenticated using user credentials. Some data stores also require connections to be made over the Secure Sockets Layer (SSL) protocol, either with or without one-way authentication. The Simba PostgreSQL ODBC Driver provides full support for these authentication protocols.

> ✎ **Note:**
>
> In this documentation, "SSL" refers to both TLS (Transport Layer Security) and SSL (Secure Sockets Layer). The driver supports TLS 1.1 and 1.2. The SSL version used for the connection is the highest version that is supported by both the driver and the server.

The driver provides a mechanism that enables you to authenticate your connection using your PostgreSQL user name and password. For detailed configuration instructions, see Creating a Data Source Name on Windows on page 9 or Creating a Data Source Name on a Non-Windows Machine on page 25.

Additionally, the driver supports SSL connections with or without one-way authentication. If the server has an SSL-enabled socket, then you can configure the driver to connect to it.

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 information about configuring SSL settings, see Configuring SSL Verification on

Windows on page 10 or Configuring SSL Verification on a Non-Windows Machine on page 30.

Driver Configuration Options

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

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

- Simba PostgreSQL ODBC Driver DSN Setup
- Additional Options
- Data Type Configuration
- SSL Options
- Logging Options

When using a connection string, 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 PostgreSQL ODBC Driver, or via the key name when using a connection string or configuring a connection from a Linux or macOS computer:

Allow Self-Signed Server Certificate

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

Description

This option specifies whether the driver allows self-signed certificates from the server.

- Enabled (1): The driver authenticates the PostgreSQL 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.

Authentication Mode

Key Name	Default Value	Required
SSLMode	prefer	No

Description

The SSL certificate verification mode to use when connecting to PostgreSQL. The following values are possible:

- **verify-full**: Connect only using SSL, a trusted certificate authority, and a server name that matches the certificate.
- **verify-ca**: Connect only using SSL and a trusted certificate authority.
- **require**: Connect only using SSL.

- **prefer**: Connect using SSL if available. Otherwise, connect without using SSL.
- **allow**: By default, connect without using SSL. If the server requires SSL connections, then use SSL.
- **disable**: Connect without using SSL.

✎ **Note:**

For information about SSL support in PostgreSQL, see "SSL Support" in the PostgreSQL Documentation: http://www.postgresql.org/docs/9.4/static/libpq-ssl.html.

Bytea As LongVarBinary

Key Name	Default Value	Required
ByteaAsLongVarBinary	0	No

Description

This option specifies the SQL data type that the driver uses to return Bytea data.

- Enabled (`1`): The driver returns Bytea columns as SQL_LONGVARBINARY data.
- Disabled (`0`): The driver returns Bytea columns as SQL_VARBINARY data.

Cache Size

Key Name	Default Value	Required
Fetch	100	Yes, if Declare/Fetch Mode is enabled.

Description

The number of rows that the driver returns when Declare/Fetch Mode is enabled. For more information, see Use Declare/Fetch on page 57.

Check Certificate Revocation

Key Name	Default Value	Required
CheckCertRevocation	Clear (0)	No

Description

This option specifies whether the driver checks to see if a certificate has been revoked while retrieving a certificate chain from the Windows Trust Store.

This option is only applicable if you are using a CA certificate from the Windows Trust Store (see Use System Trust Store on page).

- Enabled (1): The driver checks for certificate revocation while retrieving a certificate chain from the Windows Trust Store.
- Disabled (0): The driver does not check for certificate revocation while retrieving a certificate chain from the Windows Trust Store.

✎ **Note:**

This option is only available on Windows.

Custom SSL Certificate Path

Key Name	Default Value	Required
SSLCertPath	The location of the driver DLL file.	No

Description

The full path of the file containing the root certificate for verifying the server.

If this option is not set, then the driver looks in the folder that contains the driver DLL file.

Database

Key Name	Default Value	Required
Database	None	Yes

Description

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

Enable Table Types

Key Name	Default Value	Required
EnableTableTypes	Clear (0)	No

Description

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

- Clear (0): All tables returned from the data source have the generic type TABLE.
- Selected (1): The driver recognizes the following table types: TABLE, VIEW, SYSTEM TABLE, and LOCAL TEMPORARY.

Encrypt Password

Key Name	Default Value	Required
N/A	All Users Of This Machine	No

Description

This option specifies how the driver encrypts the credentials that are saved in the DSN:

- **Current User Only**: The credentials are encrypted, and can only be used by the current Windows user.
- **All Users Of This Machine**: The credentials are encrypted, but can be used by any user on the current Windows machine.

> ❗ **Important:**
>
> This option is available only when you configure a DSN using the Simba PostgreSQL ODBC Driver DSN Setup dialog box in the Windows driver. When you connect to the data store using a connection string, the driver does not encrypt your credentials.

Enforce Single Statement Mode

Key Name	Default Value	Required
EnforceSingleStatement	0	No

Description

This option specifies if the driver can have more than one active query at a time per connection.

- Enabled (1): The driver can have only one active query at a time.
- Disabled (0): The driver can have multiple active queries if the Use Multiple Statements option (the UseMultipleStatements property) is enabled. For more information, see Use Multiple Statements on page 58.

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.postgresqlodbc.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 two log files at the location you specify in the Log Path (`LogPath`) property, where *[DriverName]* is the name of the driver:

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

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 62.

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.postgresqlodbc.ini` file.

Max Bytea

Key Name	Default Value	Required
MaxBytea	255	No

Description

The maximum data length for Bytea columns.

Max LongVarChar

Key Name	Default Value	Required
MaxLongVarChar	8190	No

Description

The maximum data length for LongVarChar columns.

Max Varchar

Key Name	Default Value	Required
MaxVarchar	255	No

Description

The maximum data length for VarChar columns.

Password

Key Name	Default Value	Required
PWD OR Password	None	Yes

Description

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

Port

Key Name	Default Value	Required
Port	5432	Yes

Description

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

Proxy Port

Key Name	Default Value	Required
ProxyPort	None	Yes, if connecting through a proxy server.

Description

The number of the port that the proxy server uses to listen for client connections.

Proxy Server

Key Name	Default Value	Required
ProxyHost	None	Yes, if connecting through a proxy server.

Description

The host name or IP address of a proxy server that you want to connect through.

Retrieve Entire Result Into Memory

Key Name	Default Value	Required
N/A	Selected	No

Description

When this option is enabled, the driver returns the entire query result into memory instead of returning results in chunks or single rows at a time.

When using keys to set driver options, you can enable this option by setting the `SingleRowMode`, `UseDeclareFetch`, and `UseMultipleStatements` keys to `0`.

> ✎ **Note:**
>
> By default, the driver is configured to use this query processing mode.

Server

Key Name	Default Value	Required
Server	None	Yes

Description

The host name or IP address of the PostgreSQL server.

Show Boolean Column As String

Key Name	Default Value	Required
BoolsAsChar	Clear (0)	No

Description

This option specifies the SQL data type that the driver uses to return Boolean data.

- Enabled (1): The driver returns Boolean columns as SQL_VARCHAR data with a length of 5.
- Disabled (0): The driver returns Boolean columns as SQL_BIT data.

Single Row Mode

Key Name	Default Value	Required
SingleRowMode	Cleared (0)	No

Description

When this option is enabled (`1`), the driver uses Single Row Mode and returns query results one row at a time. Enable this option if you plan to query large results and do not want to retrieve the entire result into memory.

When using keys to set driver options, make note of the following:

- If `SingleRowMode` and `UseDeclareFetch` are both set to `0`, then the driver retrieves the entire query result into memory.
- If `UseDeclareFetch` is set to `1`, then it takes precedence over `SingleRowMode`.
- If `SingleRowMode` is set to `1` and `UseDeclareFetch` is set to `0`, then `SingleRowMode` takes precedence over `UseMultipleStatements`.

Text As LongVarChar

Key Name	Default Value	Required
TextAsLongVarchar	0	No

Description

This option specifies the SQL data type that the driver uses to return Text data. The returned data type is also affected by the Use Unicode option (the `UseUnicode` key). For more information, see Use Unicode on page 59.

- Enabled (`1`): The driver returns Text columns as SQL_LONGVARCHAR data. If the Use Unicode option (the `UseUnicode` key) is also enabled, then the driver returns SQL_WLONGVARCHAR data instead.
- Disabled (`0`): The driver returns Text columns as SQL_VARCHAR data. If the Use Unicode option (the `UseUnicode` key) is also enabled, then the driver returns SQL_WVARCHAR data instead.

Use Declare/Fetch

Key Name	Default Value	Required
UseDeclareFetch	Cleared (0)	No

Description

When this option is enabled (`1`), the driver uses Declare/Fetch Mode and returns a specific number of rows at a time. To set the number of rows, configure the Cache Size option (the `Fetch` key).

When using keys to set driver options, make note of the following:

- If `UseDeclareFetch` is set to `1`, then it takes precedence over `SingleRowMode` and `UseMultipleStatements`.
- If `UseDeclareFetch` is set to `0` and `SingleRowMode` is set to `1`, then the driver returns query results one row at a time.
- If `UseDeclareFetch` and `SingleRowMode` are both set to `0`, then the driver retrieves the entire query result into memory.

Use Multiple Statements

Key Name	Default Value	Required
UseMultipleStatements	0	No

Description

When this option is enabled (`1`), the driver can have multiple queries active on the same connection. The ODBC application may interleave calls to ODBC statements, but all queries are still sent and executed sequentially. The driver returns all the query results into memory.

When this option is disabled (`0`), the driver runs queries one at a time.

When using keys to set driver options, make note of the following:

- If `UseDeclareFetch` is set to `1`, then it takes precedence over `UseMultipleStatements`.
- If `UseDeclareFetch` is set to `0` and `SingleRowMode` is set to `1`, then `SingleRowMode` takes precedence over `UseMultipleStatements`.

Use System Trust Store

Key Name	Default Value	Required
UseSystemTrustStore	Selected (1)	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.

> ✎ **Note:**
>
> This option is only available on Windows.

Use Unicode

Key Name	Default Value	Required
UseUnicode	1	No

Description

This option specifies whether the driver returns PostgreSQL data as Unicode or regular SQL types.

When this option is enabled (1), the driver returns data as Unicode character types:

- SQL_WCHAR is returned instead of SQL_CHAR.
- SQL_WVARCHAR is returned instead of SQL_VARCHAR.
- SQL_WLONGVARCHAR is returned instead of SQL_LONGVARCHAR.

When this option is disabled (0), the driver returns data as regular SQL types:

- SQL_CHAR is returned instead of SQL_WCHAR.
- SQL_VARCHAR is returned instead of SQL_WVARCHAR.
- SQL_LONGVARCHAR is returned instead of SQL_WLONGVARCHAR.

For detailed information about how the driver returns PostgreSQL data as SQL types, see Data Types on page 40.

User

Key Name	Default Value	Required
UID OR Username	None	Yes

Description

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

If you are using keys to set driver options, UID takes precedence over Username.

Configuration Options Having Only Key Names

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

Driver

Key Name	Default Value	Required
Driver	Simba PostgreSQL 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 PostgreSQL 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.

KeepAlive

Key Name	Default Value	Required
KeepAlive	1	No

Description

When this option is enabled (`1`), the driver uses TCP keepalives to prevent connections from timing out.

When this option is disabled (`0`), the driver does not use TCP keepalives.

KeepAliveCount

Key Name	Default Value	Required
KeepAliveCount	0	No

Description

The number of TCP keepalive packets that can be lost before the connection is considered broken.

When this key is set to `0`, the driver uses the system default for this setting.

KeepAliveInterval

Key Name	Default Value	Required
KeepAliveInterval	0	No

Description

The number of seconds between each TCP keepalive retransmission.

When this key is set to `0`, the driver uses the system default for this setting.

KeepAliveTime

Key Name	Default Value	Required
KeepAliveTime	0	No

Description

The number of seconds of inactivity before the driver sends a TCP keepalive packet.

When this key is set to `0`, the driver uses the system default for this setting.

Locale

Key Name	Default Value	Required
Locale	en-US	No

Description

The locale to use for error messages.

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_PostgreSQLODBC_driver.log` and `jdoe_7836_`

`SimbaPostgreSQLODBCDriver_connection_ [Number].log,` where
[Number] is a number that identifies each connection-specific log file.

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

Third-Party Trademarks

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, 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.

PostgreSQL is a trademark or registered trademark of The PostgreSQL Global Development Group or its subsidiaries in Canada, the 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 NOINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

ICU License - ICU 1.8.1 and later

COPYRIGHT AND PERMISSION NOTICE

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

All rights reserved.

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

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

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

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

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

OpenSSL License

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

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

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

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

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

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

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

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

Original SSLeay License

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

All rights reserved.

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

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

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

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

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

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

The word 'cryptographic' can be left out if the rouines 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.]

PostgreSQL Database Management System License

(formerly known as Postgres, then as Postgres95)

Portions Copyright (c) 1996-2015, The PostgreSQL Global Development Group

Portions Copyright (c) 1994, The Regents of the University of California

Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE

UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

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

UPX License

-----BEGIN PGP SIGNED MESSAGE-----

```
ooooo      ooo oooooooo.    ooooooo   ooooo
`888'      `8' `888   `Y88. `8888    d8'
 888        8   888   .d88'  Y888..8P
 888        8   888ooo88P'    `8888'
 888        8   888          .8PY888.
 `88.      .8'  888         d8'  `888b
   `YbodP'    o8880      o8880  o88888o
```

The Ultimate Packer for eXecutables
Copyright (c) 1996-2000 Markus Oberhumer & Laszlo Molnar
http://wildsau.idv.uni-linz.ac.at/mfx/upx.html
http://www.nexus.hu/upx
http://upx.tsx.org

PLEASE CAREFULLY READ THIS LICENSE AGREEMENT, ESPECIALLY IF YOU
PLAN TO MODIFY THE UPX SOURCE CODE OR USE A MODIFIED UPX
VERSION.

ABSTRACT

========

UPX and UCL are copyrighted software distributed under the terms of the GNU
General Public License (hereinafter the "GPL").

The stub which is imbedded in each UPX compressed program is part of UPX and
UCL, and contains code that is under our copyright. The terms of the GNU General
Public License still apply as compressing a program is a special form of linking with
our stub.

As a special exception we grant the free usage of UPX for all executables, including
commercial programs. See below for details and restrictions.

COPYRIGHT

=========

UPX and UCL are copyrighted software. All rights remain with the authors.

UPX is Copyright (C) 1996-2000 Markus Franz Xaver Johannes Oberhumer
UPX is Copyright (C) 1996-2000 Laszlo Molnar

UCL is Copyright (C) 1996-2000 Markus Franz Xaver Johannes Oberhumer

GNU GENERAL PUBLIC LICENSE

=========================

UPX and the UCL library are free software; you can redistribute them and/or modify
them under the terms of the GNU General Public License as published by the Free

Software Foundation; either version 2 of the License, or (at your option) any later version.

UPX and UCL are distributed in the hope that they will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; see the file COPYING.

SPECIAL EXCEPTION FOR COMPRESSED EXECUTABLES

==

The stub which is imbedded in each UPX compressed program is part of UPX and UCL, and contains code that is under our copyright. The terms of the GNU General Public License still apply as compressing a program is a special form of linking with our stub.

Hereby Markus F.X.J. Oberhumer and Laszlo Molnar grant you special permission to freely use and distribute all UPX compressed programs (including commercial ones), subject to the following restrictions:

1. You must compress your program with a completely unmodified UPX version; either with our precompiled version, or (at your option) with a self compiled version of the unmodified UPX sources as distributed by us.
2. This also implies that the UPX stub must be completely unmodfied, i.e. the stub imbedded in your compressed program must be byte-identical to the stub that is produced by the official unmodified UPX version.
3. The decompressor and any other code from the stub must exclusively get used by the unmodified UPX stub for decompressing your program at program startup. No portion of the stub may get read, copied, called or otherwise get used or accessed by your program.

ANNOTATIONS

===========

- You can use a modified UPX version or modified UPX stub only for programs that are compatible with the GNU General Public License.

- We grant you special permission to freely use and distribute all UPX compressed programs. But any modification of the UPX stub (such as, but not limited to, removing our copyright string or making your program non-decompressible) will immediately revoke your right to use and distribute a UPX compressed program.

- UPX is not a software protection tool; by requiring that you use the unmodified UPX version for your proprietary programs we make sure that any user can decompress your program. This protects both you and your users as nobody can hide malicious code - any program that cannot be decompressed is highly suspicious by definition.

- You can integrate all or part of UPX and UCL into projects that are compatible with the GNU GPL, but obviously you cannot grant any special exceptions beyond the GPL for our code in your project.

- We want to actively support manufacturers of virus scanners and similar security software. Please contact us if you would like to incorporate parts of UPX or UCL into such a product.

Markus F.X.J. Oberhumer	Laszlo Molnar
markus.oberhumer@jk.uni-linz.ac.at	ml1050@cdata.tvnet.hu

Linz, Austria, 25 Feb 2000

-----BEGIN PGP SIGNATURE-----

Version: 2.6.3ia
Charset: noconv

iQCVAwUBOLaLS210fyLu8beJAQFYVAP/ShzENWKLTvedLCjZbDcwaBEHfUVcrG
MI
wE7frMkbWT2zmkdv9hW90WmjMhOBu7yhUplvN8BKOtLiolEnZmLCYu8AGCwr5wB
f
dfLoCIxnzfTtgQv5axF1awp4RwCUH3hf4cDrOVqmAsWXKPHtm4hx96jF6L4oHhjx
OO03+ojZdO8=
=CS52

-----END PGP SIGNATURE-----