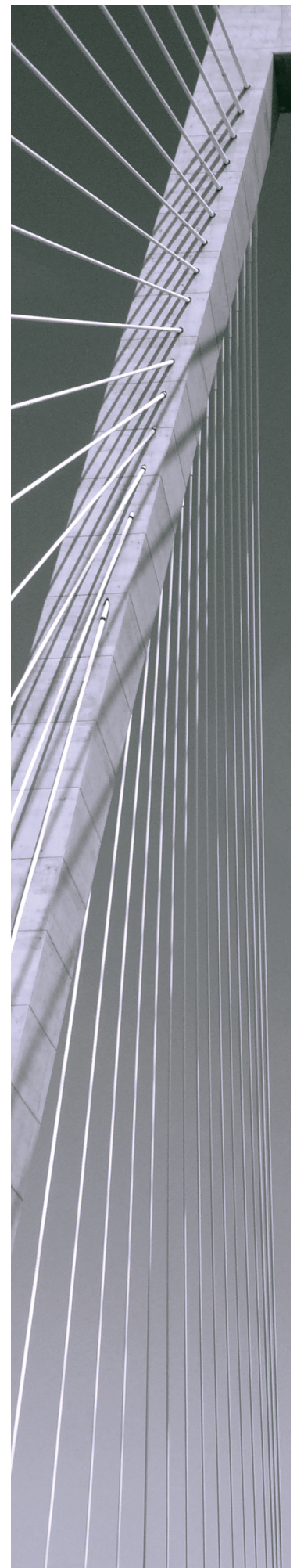# Simba Hive ODBC Driver with SQL Connector

Installation and Configuration Guide

Simba Technologies Inc.

Version 2.6.9

July 17, 2020

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

Audience

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

Knowledge Prerequisites

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

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

The Simba Hive ODBC Driver is used for direct SQL and HiveQL access to Apache Hadoop / Hive distributions, enabling Business Intelligence (BI), analytics, and reporting on Hadoop / Hive-based data. The driver efficiently transforms an application's SQL query into the equivalent form in HiveQL, which is a subset of SQL-92. If an application is Hive-aware, then the driver is configurable to pass the query through to the database for processing. The driver interrogates Hive to obtain schema information to present to a SQL-based application. Queries, including joins, are translated from SQL to HiveQL. For more information about the differences between HiveQL and SQL, see SQL Connector for HiveQL on page 72.

The Simba Hive ODBC Driver complies with the ODBC 3.80 data standard and adds important functionality such as Unicode and 32- and 64-bit support for high-performance computing environments.

ODBC is one of the most established and widely supported APIs for connecting to and working with databases. At the heart of the technology is the ODBC driver, which connects an application to the database. For more information about ODBC, see *Data Access Standards* on the Simba Technologies website: https://www.simba.com/resources/data-access-standards-glossary. For complete information about the ODBC specification, see the *ODBC API Reference* from the Microsoft documentation: https://docs.microsoft.com/en-us/sql/odbc/reference/syntax/odbc-api-reference.

The Simba Hive ODBC Driver is available for Microsoft® Windows®, Linux, Solaris, AIX, and macOS platforms.

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

> ✎ **Note:**
>
> For basic configuration instructions that allow you to quickly set up the Windows driver so that you can evaluate and use it, see the *Simba ODBC Drivers Quick Start Guide for Windows*. The Quick Start Guide also explains how to use the driver in various applications.

Windows Driver

Windows System Requirements

The Simba Hive ODBC Driver supports Apache Hive versions 1.0.0 through 3.1.0.

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

- Administrator rights on your machine.
- A machine that meets the following system requirements:
    - One of the following operating systems:
        - Windows 10, 8.1, or 7 SP1
        - Windows Server 2019, 2016, 2012, or 2008 R2 SP1
    - 100 MB of available disk space

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

Installing the Driver on Windows

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

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

- `SimbaHiveODBC32.msi` for 32-bit applications
- `SimbaHiveODBC64.msi` for 64-bit applications

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

To install the Simba Hive ODBC Driver on Windows:

1. Depending on the bitness of your client application, double-click to run **SimbaHiveODBC32.msi** or **SimbaHiveODBC64.msi**.

2. Click **Next**.

3. Select the check box to accept the terms of the License Agreement if you agree, and then click **Next**.

4. To change the installation location, click **Change**, then browse to the desired folder, and then click **OK**. To accept the installation location, click **Next**.

5. Click **Install**.

6. When the installation completes, click **Finish**.

7. If you received a license file through email, then copy the license file into the `\lib` subfolder of the installation folder you selected above. You must have Administrator privileges when changing the contents of this folder.

Creating a Data Source Name on Windows

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

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

The following instructions describe how to create a DSN. For information about specifying settings in a connection string, see Using a Connection String on page 68. For information about driver-wide settings, see Configuring a DSN-less Connection on Windows on page 12.

To create a Data Source Name on Windows:

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

   > ✎ **Note:**
   >
   > Make sure to select the ODBC Data Source Administrator that has the same bitness as the client application that you are using to connect to Hive.

2. In the ODBC Data Source Administrator, click the **Drivers** tab, and then scroll down as needed to confirm that the Simba Hive 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 Hive ODBC Driver** and then click **Finish**. The Simba Hive ODBC Driver DSN Setup dialog box opens.
6. In the **Data Source Name** field, type a name for your DSN.
7. Optionally, in the **Description** field, type relevant details about the DSN.
8. In the **Hive Server Type** drop-down list, select **Hive Server 1** or **Hive Server 2**.

> ✎ **Note:**
>
> If you are connecting through Apache ZooKeeper, then **Hive Server 1** is not supported.

9. Specify whether the driver uses the ZooKeeper service when connecting to Hive, and provide the necessary connection information:

- To connect to Hive without using the Apache ZooKeeper service, do the following:
    a. From the **Service Discovery Mode** drop-down list, select **No Service Discovery**.
    b. In the **Host(s)** field, type the IP address or host name of the Hive server.
    c. In the **Port** field, type the number of the TCP port that the Hive server uses to listen for client connections.
- Or, to discover Hive Server 2 services via the ZooKeeper service, do the following:
    a. From the **Service Discovery Mode** drop-down list, select **ZooKeeper**.
    b. In the **Host(s)** field, type a comma-separated list of ZooKeeper servers. Use the following format, where *[ZK_Host]* is the IP address or host name of the ZooKeeper server and *[ZK_Port]* is the number of the TCP port that the ZooKeeper server uses to listen for client connections:

```
[ZK_Host1]:[ZK_Port1],[ZK_Host2]:[ZK_Port2]
```

      c. In the **ZooKeeper Namespace** field, type the namespace on ZooKeeper under which Hive Server 2 znodes are added.

10. In the **Database** field, type the name of the database schema to use when a schema is not explicitly specified in a query.

> ✎ **Note:**
>
> You can still issue queries on other schemas by explicitly specifying the schema in the query. To inspect your databases and determine the appropriate schema to use, type the `show databases` command at the Hive command prompt.

11. In the Authentication area, configure authentication as needed. For more information, see Configuring Authentication on Windows on page 15.

> ✎ **Note:**
>
> Hive Server 1 does not support authentication. Most default configurations of Hive Server 2 require User Name authentication. To verify the authentication mechanism that you need to use for your connection, check the configuration of your Hadoop / Hive distribution. For more information, see Authentication Mechanisms on page 66.

12. Optionally, if the operations against Hive are to be done on behalf of a user that is different than the authenticated user for the connection, type the name of the user to be delegated in the **Delegation UID** field. For more information, see Delegating Authentication to a Specific User on page 19.

> ✎ **Note:**
>
> This option is applicable only when connecting to a Hive Server 2 instance that supports this feature.

13. In the **Thrift Transport** drop-down list, select the transport protocol to use in the Thrift layer.

> ✎ **Note:**
>
> For information about how to determine which Thrift transport protocols your Hive server supports, see Authentication Mechanisms on page 66.

14. If the Thrift Transport option is set to HTTP, then to configure HTTP options such as custom headers, click **HTTP Options**. For more information, see Configuring HTTP Options on Windows on page 22.

15. To configure client-server verification over SSL, click **SSL Options**. For more information, see Configuring SSL Verification on Windows on page 23.

> ✎ **Note:**
>
> If you selected User Name as the authentication mechanism, SSL is not available.

16. To configure advanced driver options, click **Advanced Options**. For more information, see Configuring Advanced Options on Windows on page 20.

17. To configure server-side properties, click **Advanced Options** and then click **Server Side Properties**. For more information, see Configuring Server-Side Properties on Windows on page 25.

18. To configure logging behavior for the driver, click **Logging Options**. For more information, see Configuring Logging Options on Windows on page 26.

19. 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 Hive ODBC Driver DSN Setup dialog box are correct. Contact your Hive server administrator as needed.

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

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

Configuring a DSN-less Connection on Windows

Some client applications provide support for connecting to a data source using a driver without a Data Source Name (DSN). To configure a DSN-less connection, you can use a connection string or the Simba Hive ODBC Driver Configuration tool that is installed with the Simba Hive ODBC Driver. Settings in a connection string apply only when you connect to Hive using that particular string, while settings in the driver configuration tool apply to every connection that uses the Simba Hive ODBC Driver.

The following section explains how to use the driver configuration tool. For information about using connection strings, see Using a Connection String on page 68.

> ✎ **Note:**
>
> - Settings in the connection string take precedence over settings in the DSN, and settings in the DSN take precedence over driver-wide settings.
> - The drop-down lists in the driver configuration tool only display one option at a time. Use the scroll arrows on the right side of the drop-down list to view and select other options.

To configure a DSN-less connection using the driver configuration tool:

1. Choose one:

   - If you are using Windows 7 or earlier, click **Start** 🌐 **> All Programs > Simba Hive ODBC Driver 2.6 > Driver Configuration**.
   - Or, if you are using Windows 8 or later, click the arrow button at the bottom of the Start screen, and then click **Simba Hive ODBC Driver 2.6 > Driver Configuration**.

   > ✎ **Note:**
   >
   > Make sure to select the Driver Configuration Tool that has the same bitness as the client application that you are using to connect to Hive.

2. If you are prompted for administrator permission to make modifications to the machine, click **OK**.

   > ✎ **Note:**
   >
   > You must have administrator access to the machine to run this application because it makes changes to the registry.

3. In the **Hive Server Type** drop-down list, select **Hive Server 1** or **Hive Server 2**.

   > ✎ **Note:**
   >
   > If you are connecting through Apache ZooKeeper, then **Hive Server 1** is not supported.

4. Specify whether the driver uses the ZooKeeper service when connecting to Hive:

   - To connect to Hive without using the Apache ZooKeeper service, from the **Service Discovery Mode** drop-down list, select **No Service Discovery**.
   - Or, to discover Hive Server 2 services via the ZooKeeper service, do the following:

     a. From the **Service Discovery Mode** drop-down list, select **ZooKeeper**.

     b. In the **Host(s)** field, type a comma-separated list of ZooKeeper servers. Use the following format, where *[ZK_Host]* is the IP address or host name of the ZooKeeper server and *[ZK_Port]* is the number of the TCP port that the ZooKeeper server uses to listen for client connections:

     ```
     [ZK_Host1]:[ZK_Port1],[ZK_Host2]:[ZK_Port2]
     ```

     c. In the **ZooKeeper Namespace** field, type the namespace on ZooKeeper under which Hive Server 2 znodes are added.

5. In the Authentication area, configure authentication as needed. For more information, see Configuring Authentication on Windows on page 15.

> ✎ **Note:**
>
> Hive Server 1 does not support authentication. Most default configurations of Hive Server 2 require User Name authentication. To verify the authentication mechanism that you need to use for your connection, check the configuration of your Hadoop / Hive distribution. For more information, see Authentication Mechanisms on page 66.

6. Optionally, if the operations against Hive are to be done on behalf of a user that is different than the authenticated user for the connection, then in the **Delegation UID** field, type the name of the user to be delegated. For more information, see Delegating Authentication to a Specific User on page 19.

> ✎ **Note:**
>
> This option is applicable only when connecting to a Hive Server 2 instance that supports this feature.

7. In the **Thrift Transport** drop-down list, select the transport protocol to use in the Thrift layer.

> ✎ **Note:**
>
> For information about how to determine which Thrift transport protocols your Hive server supports, see Authentication Mechanisms on page 66.

8. If the Thrift Transport option is set to HTTP, then to configure HTTP options such as custom headers, click **HTTP Options**. For more information, see Configuring HTTP Options on Windows on page 22.

9. To configure client-server verification over SSL, click **SSL Options**. For more information, see Configuring SSL Verification on Windows on page 23.

> ✎ **Note:**
>
> If you selected User Name as the authentication mechanism, SSL is not available.

10. To configure advanced options, click **Advanced Options**. For more information, see Configuring Advanced Options on Windows on page 20.

11. To configure server-side properties, click **Advanced Options** and then click **Server Side Properties**. For more information, see Configuring Server-Side Properties on Windows on page 25.

12. To save your settings and close the Simba Hive ODBC Driver Configuration tool, click **OK**.

Configuring Authentication on Windows

Some Hive Server 2 instances are configured to require authentication for access. To connect to a Hive server, you must configure the Simba Hive ODBC Driver to use the authentication mechanism that matches the access requirements of the server and provides the necessary credentials.

For information about how to determine the type of authentication your Hive server requires, see Authentication Mechanisms on page 66.

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

If cookie-based authentication is enabled in your Hive Server 2 database, you can specify a list of authentication cookies in the `HTTPAuthCookies` connection property. In this case, the driver authenticates the connection once based on the provided authentication credentials. It then uses the cookie generated by the server for each subsequent request in the same connection. For more information, see HTTPAuthCookies on page 107.

> ✎ **Note:**
>
> On Windows, the `HTTPAuthCookies` property must be set in a connection string.

Using No Authentication

When connecting to a Hive server of type Hive Server 1, you must use No Authentication. When you use No Authentication, Binary is the only Thrift transport protocol that is supported.

To configure a connection without authentication:

1. Choose one:
   - To access authentication options for a DSN, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, and then click **Configure**.
   - Or, to access authentication options for a DSN-less connection, open the Simba Hive ODBC Driver Configuration tool.
2. From the **Mechanism** drop-down list, select **No Authentication**.

3. If the Hive server is configured to use SSL, then click **SSL Options** to configure SSL for the connection. For more information, see Configuring SSL Verification on Windows on page 23.

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

Using Kerberos

This authentication mechanism is available only for Hive Server 2 on non-HDInsight distributions. When you use Kerberos authentication, the Binary transport protocol is not supported.

If the Use Only SSPI advanced option is disabled, then Kerberos must be installed and configured before you can use this authentication mechanism. For information about configuring Kerberos on your machine, see Configuring Kerberos Authentication for Windows on page 28. For information about setting the Use Only SSPI advanced option, see Configuring Advanced Options on Windows on page 20.

To configure Kerberos authentication:

1. Choose one:
   - To access authentication options for a DSN, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, and then click **Configure**.
   - Or, to access authentication options for a DSN-less connection, open the Simba Hive ODBC Driver Configuration tool.

2. From the **Mechanism** drop-down list, select **Kerberos**.

3. Choose one:
   - To use the default realm defined in your Kerberos setup, leave the **Realm** field empty.
   - Or, if your Kerberos setup does not define a default realm or if the realm of your Hive Server 2 host is not the default, then, in the **Realm** field, type the Kerberos realm of the Hive Server 2.

4. In the **Host FQDN** field, type the fully qualified domain name of the Hive Server 2 host.

> ✎ **Note:**
>
> To use the Hive server host name as the fully qualified domain name for Kerberos authentication, in the **Host FQDN** field, type **_HOST**.

5. In the **Service Name** field, type the service name of the Hive server.

6. Optionally, if you are using MIT Kerberos and a Kerberos realm is specified in the **Realm** field, then choose one:

- To have the Kerberos layer canonicalize the server's service principal name, leave the **Canonicalize Principal FQDN** check box selected.
- Or, to prevent the Kerberos layer from canonicalizing the server's service principal name, clear the **Canonicalize Principal FQDN** check box.

7. To allow the driver to pass your credentials directly to the server for use in authentication, select **Delegate Kerberos Credentials**.

8. From the **Thrift Transport** drop-down list, select the transport protocol to use in the Thrift layer.

> ❗ **Important:**
>
> When using this authentication mechanism, the Binary transport protocol is not supported.

9. If the Hive server is configured to use SSL, then click **SSL Options** to configure SSL for the connection. For more information, see Configuring SSL Verification on Windows on page 23.

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

Using User Name

This authentication mechanism requires a user name but not a password. The user name labels the session, facilitating database tracking.

This authentication mechanism is available only for Hive Server 2 on non-HDInsight distributions. Most default configurations of Hive Server 2 require User Name authentication. When you use User Name authentication, SSL is not supported and SASL is the only Thrift transport protocol available.

To configure User Name authentication:

1. Choose one:
   - To access authentication options for a DSN, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, and then click **Configure**.
   - Or, to access authentication options for a DSN-less connection, open the Simba Hive ODBC Driver Configuration tool.

2. From the **Mechanism** drop-down list, select **User Name**.

3. In the **User Name** field, type an appropriate user name for accessing the Hive server.

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

Using User Name And Password

This authentication mechanism requires a user name and a password.

This authentication mechanism is available only for Hive Server 2 on non-HDInsight distributions.

To configure User Name And Password authentication:

1. Choose one:
    - To access authentication options for a DSN, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, and then click **Configure**.
    - Or, to access authentication options for a DSN-less connection, open the Simba Hive ODBC Driver Configuration tool.
2. From the **Mechanism** drop-down list, select **User Name And Password**.
3. In the **User Name** field, type an appropriate user name for accessing the Hive server.
4. In the **Password** field, type the password corresponding to the user name you typed above.
5. To save the password, select the **Save Password (Encrypted)** check box.

> **! Important:**
>
> The password is obscured, that is, not saved in plain text. However, it is still possible for the encrypted password to be copied and used.

6. From the **Thrift Transport** drop-down list, select the transport protocol to use in the Thrift layer.
7. If the Hive server is configured to use SSL, then click **SSL Options** to configure SSL for the connection. For more information, see Configuring SSL Verification on Windows on page 23.
8. To save your settings and close the dialog box, click **OK**.

Using Windows Azure HDInsight Service

This authentication mechanism is available only for Hive Server 2 on HDInsight distributions. When you use this authentication mechanism, you must enable SSL and HTTP is the only Thrift transport protocol available.

To configure a connection to a Hive server on Windows Azure HDInsight Service:

1. Choose one:
    - To access authentication options for a DSN, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, and then click **Configure**.

- Or, to access authentication options for a DSN-less connection, open the Simba Hive ODBC Driver Configuration tool.

2. From the **Mechanism** drop-down list, select **Windows Azure HDInsight Service**.

3. In the **User Name** field, type an appropriate user name for accessing the Hive server.

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

5. To save the password, select the **Save Password (Encrypted)** check box.

> ❗ **Important:**
>
> The password is obscured, that is, not saved in plain text. However, it is still possible for the encrypted password to be copied and used.

6. Click **HTTP Options**, and in the **HTTP Path** field, type the partial URL corresponding to the Hive server. Click **OK** to save your HTTP settings and close the dialog box.

> ✎ **Note:**
>
> If necessary, you can create custom HTTP headers. For more information, see Configuring HTTP Options on Windows on page 22.

7. Click **SSL Options** and configure SSL settings as needed. For more information, see Configuring SSL Verification on Windows on page 23.

8. Click **OK** to save your SSL configuration and close the dialog box, and then click **OK** to save your authentication settings and close the dialog box.

Delegating Authentication to a Specific User

Some Hive Server 2 instances support the ability to delegate all operations against Hive to the specified user, rather than to the authenticated user for the connection.

To delegate all operations to a specified user:

1. Choose one:
   - To access authentication options for a DSN, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, and then click **Configure**.
   - Or, to access authentication options for a DSN-less connection, open the Simba Hive ODBC Driver Configuration tool.

2. In the **Delegation UID** field, type the name of the user to be delegated.

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

If the server returns an error message such as **Failed to validate proxy privilege of [RealUser] for [DelegationUID]**, you may need to modify the server's `core-site.xml` configuration file, as follows:

1. In the server's `core-site.xml` configuration file, add the following properties:

   ```
   hadoop.proxyuser.[RealUser].groups=*
   hadoop.proxyuser.[RealUser].hosts=*
   ```

   Where *[RealUser]* is the authenticated user for the connection.

2. If you are using Kerberos authentication, then in the server's `core-site.xml` configuration file, add the following properties:

   ```
   hadoop.proxyuser.[Principal].groups=*
   hadoop.proxyuser.[Principal].hosts=*
   ```

   Where *[Principal]* is the primary Kerberos principal user. For example, if the primary Kerberos principal user is `kerbuser@example.com`, replace *[Principal]* with `kerbuser`.

For more information on resolving this error, see your Hive Server documentation.

Configuring Advanced Options on Windows

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

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

To configure advanced options on Windows:

1. Choose one:
   - To access advanced 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 **Advanced Options**.
   - Or, to access advanced options for a DSN-less connection, open the Simba Hive ODBC Driver Configuration tool, and then click **Advanced Options**.
2. To disable the SQL Connector feature, select the **Use Native Query** check box.

> **! Important:**
>
> - When this option is enabled, the driver cannot execute parameterized queries.
> - By default, the driver applies transformations to the queries emitted by an application to convert the queries into an equivalent form in HiveQL. If the application is Hive-aware and already emits HiveQL, then turning off the translation avoids the additional overhead of query transformation.

3. To defer query execution to SQLExecute, select the **Fast SQLPrepare** check box.

4. To allow driver-wide configurations to take precedence over connection and DSN settings, select the **Driver Config Take Precedence** check box.

5. To use the asynchronous version of the API call against Hive for executing a query, select the **Use Async Exec** check box.

6. To retrieve table names from the database by using the SHOW TABLES query, select the **Get Tables With Query** check box.

> **✎ Note:**
>
> This option is applicable only when connecting to Hive Server 2.

7. To enable the driver to return SQL_WVARCHAR instead of SQL_VARCHAR for STRING and VARCHAR columns, and SQL_WCHAR instead of SQL_CHAR for CHAR columns, select the Unicode SQL Character Types check box.

8. To enable the driver to return the hive_system table for catalog function calls such as SQLTables and SQLColumns, select the **Show System Table** check box.

9. To specify which mechanism the driver uses by default to handle Kerberos authentication, do one of the following:

   - To use the SSPI plugin by default, select the **Use Only SSPI** check box.
   - To use MIT Kerberos by default and only use the SSPI plugin if the GSSAPI library is not available, clear the **Use Only SSPI** check box.

10. To enable the driver to automatically open a new session when the existing session is no longer valid, select the **Invalid Session Auto Recover** check box.

> **✎ Note:**
>
> This option is applicable only when connecting to Hive Server 2.

11. To have the driver automatically attempt to reconnect to the server if communications are lost, select **Enable Auto Reconnect**.

12. In the **Rows Fetched Per Block** field, type the number of rows to be fetched per block.

13. In the **Max Bytes Per Fetch Request** field, type the maximum number of bytes to be fetched.

> ✎ **Note:**
>
> - This option is applicable only when connecting to a server that supports result set data serialized in arrow format.
> - The value must be specified in one of the following:
>   ◦ B (bytes)
>   ◦ KB (kilobytes)
>   ◦ MB (megabytes)
>   ◦ GB (gigabytes)
>
>   By default, the file size is in B (bytes).

14. In the **Default String Column Length** field, type the maximum data length for STRING columns.

15. In the **Binary Column Length** field, type the maximum data length for BINARY columns.

16. In the **Decimal Column Scale** field, type the maximum number of digits to the right of the decimal point for numeric data types.

17. In the **Async Exec Poll Interval** field, type the time in milliseconds between each poll for the query execution status.

> ✎ **Note:**
>
> This option is applicable only to HDInsight clusters.

18. In the **Socket Timeout** field, type the number of seconds that an operation can remain idle before it is closed.

> ✎ **Note:**
>
> This option is applicable only when asynchronous query execution is being used against Hive Server 2 instances.

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

Configuring HTTP Options on Windows

You can configure options such as custom headers when using the HTTP transport protocol in the Thrift layer. For information about how to determine if your Hive server supports the HTTP transport protocol, see Authentication Mechanisms on page 66.

The following instructions describe how to configure HTTP options in a DSN and in the driver configuration tool. You can specify the connection settings described below

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

To configure HTTP options on Windows:

1. Choose one:
   - If you are configuring HTTP for a DSN, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, and then make sure that the Thrift Transport option is set to **HTTP**.
   - Or, if you are configuring HTTP for a DSN-less connection, open the Simba Hive ODBC Driver Configuration tool and then make sure that the Thrift Transport option is set to **HTTP**.
2. To access HTTP options, click **HTTP Options**.

> ✎ **Note:**
>
> The HTTP options are available only when the Thrift Transport option is set to HTTP.

3. In the **HTTP Path** field, type the partial URL corresponding to the Hive server.
4. To create a custom HTTP header, click **Add**, then type appropriate values in the **Key** and **Value** fields, and then click **OK**.
5. To edit a custom HTTP header, select the header from the list, then click **Edit**, then update the **Key** and **Value** fields as needed, and then click **OK**.
6. To delete a custom HTTP header, select the header from the list, and then click **Remove**. In the confirmation dialog box, click **Yes**.
7. To save your settings and close the HTTP Properties dialog box, click **OK**.

Configuring SSL Verification on Windows

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

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

✎ **Note:**

If you selected User Name as the authentication mechanism, SSL is not available.

To configure SSL verification on Windows:

1. Choose one:
   - To access 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**.
   - Or, to access advanced options for a DSN-less connection, open the Simba Hive ODBC Driver Configuration tool, and then click **SSL Options**.

2. Select the **Enable SSL** check box.

3. To allow authentication using self-signed certificates that have not been added to the list of trusted certificates, select the **Allow Self-signed Server Certificate** check box.

4. To allow the common name of a CA-issued SSL certificate to not match the host name of the Hive server, select the **Allow Common Name Host Name Mismatch** check box.

5. To specify the CA certificates that you want to use to verify the server, do one of the following:
   - To verify the server using the trusted CA certificates from a specific `.pem` file, specify the full path to the file in the **Trusted Certificates** field and clear the **Use System Trust Store** check box.
   - Or, to use the trusted CA certificates `.pem` file that is installed with the driver, leave the **Trusted Certificates** field empty, and clear the **Use System Trust Store** check box.
   - Or, to use the Windows trust store, select the **Use System Trust Store** check box.

   ! **Important:**
   - If you are using the Windows trust store, make sure to import the trusted CA certificates into the trust store.
   - If the trusted CA supports certificate revocation, select the **Check Certificate Revocation** check box.

6. From the **Minimum TLS Version** drop-down list, select the minimum version of TLS to use when connecting to your data store.

7. To configure two-way SSL verification, select the **Two-Way SSL** check box and then do the following:
   a. In the **Client Certificate File** field, specify the full path of the PEM file containing the client's certificate.

b. In the **Client Private Key File** field, specify the full path of the file containing the client's private key.

c. If the private key file is protected with a password, type the password in the **Client Private Key Password** field. To save the password, select the **Save Password (Encrypted)** check box.

> ❗ **Important:**
>
> The password is obscured, that is, not saved in plain text. However, it is still possible for the encrypted password to be copied and used.

8. To save your settings and close the SSL Options dialog box, click **OK**.

Configuring Server-Side Properties on Windows

You can use the driver to apply configuration properties to the Hive server.

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

To configure server-side properties on Windows:

1. Choose one:
   - To configure server-side properties for a DSN, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, then click **Advanced Options**, and then click **Server Side Properties**.
   - Or, to configure server-side properties for a DSN-less connection, open the Simba Hive ODBC Driver Configuration tool, then click **Advanced Options**, and then click **Server Side Properties**.
2. To create a server-side property, click **Add**, then type appropriate values in the **Key** and **Value** fields, and then click **OK**. For example, to set the value of the `mapreduce.job.queuename` property to `myQueue`, type **mapreduce.job.queuename** in the **Key** field and then type **myQueue** in the **Value** field.

> ✎ **Note:**
>
> For a list of all Hadoop and Hive server-side properties that your implementation supports, type `set -v` at the Hive CLI command line or Beeline. You can also execute the `set -v` query after connecting using the driver.

3. To edit a server-side property, select the property from the list, then click **Edit**, then update the **Key** and **Value** fields as needed, and then click **OK**.

4. To delete a server-side property, select the property from the list, and then click **Remove**. In the confirmation dialog box, click **Yes**.

5. To configure the driver to convert server-side property key names to all lower-case characters, select the **Convert Key Name To Lower Case** check box.

6. To change the method that the driver uses to apply server-side properties, do one of the following:

   - To configure the driver to apply each server-side property by executing a query when opening a session to the Hive server, select the **Apply Server Side Properties With Queries** check box.

   - Or, to configure the driver to use a more efficient method for applying server-side properties that does not involve additional network round-tripping, clear the **Apply Server Side Properties With Queries** check box.

> ✎ **Note:**
>
> The more efficient method is not available for Hive Server 1, and it might not be compatible with some Hive Server 2 builds. If the server-side properties do not take effect when the check box is clear, then select the check box.

7. To save your settings and close the Server Side Properties dialog box, click **OK**.

Configuring Logging Options on Windows

To help troubleshoot issues, you can enable logging. In addition to functionality provided in the Simba Hive 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 Hive 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. If requested by Technical Support, type the name of the component for which to log messages in the **Log Namespace** field. Otherwise, do not type a value in the field.
5. In the **Max Number Files** field, type the maximum number of log files to keep.

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

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

> ✎ **Note:**
>
> After the maximum file size is reached, the driver creates a new file and continues logging.

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

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

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

If you enable the `UseLogPrefix` connection property, the driver prefixes the log file name with the user name associated with the connection and the process ID of the application through which the connection is made. For more information, see UseLogPrefix on page 108.

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.

Configuring Kerberos Authentication for Windows

Active Directory

The Simba Hive ODBC Driver supports Active Directory Kerberos on Windows. There are two prerequisites for using Active Directory Kerberos on Windows:

- MIT Kerberos is not installed on the client Windows machine.
- The MIT Kerberos Hadoop realm has been configured to trust the Active Directory realm so that users in the Active Directory realm can access services in the MIT Kerberos Hadoop realm.

MIT Kerberos

Downloading and Installing MIT Kerberos for Windows 4.0.1

For information about Kerberos and download links for the installer, see the MIT Kerberos website: http://web.mit.edu/kerberos/.

To download and install MIT Kerberos for Windows 4.0.1:

1. Download the appropriate Kerberos installer:
   - For a 64-bit machine, use the following download link from the MIT Kerberos website: http://web.mit.edu/kerberos/dist/kfw/4.0/kfw-4.0.1-amd64.msi.

- For a 32-bit machine, use the following download link from the MIT Kerberos website: http://web.mit.edu/kerberos/dist/kfw/4.0/kfw-4.0.1-i386.msi.

> ✎ **Note:**
>
> The 64-bit installer includes both 32-bit and 64-bit libraries. The 32-bit installer includes 32-bit libraries only.

2. To run the installer, double-click the `.msi` file that you downloaded above.
3. Follow the instructions in the installer to complete the installation process.
4. When the installation completes, click **Finish**.

Setting Up the Kerberos Configuration File

Settings for Kerberos are specified through a configuration file. You can set up the configuration file as an `.ini` file in the default location, which is the `C:\ProgramData\MIT\Kerberos5` directory, or as a `.conf` file in a custom location.

Normally, the `C:\ProgramData\MIT\Kerberos5` directory is hidden. For information about viewing and using this hidden directory, refer to Microsoft Windows documentation.

> ✎ **Note:**
>
> For more information on configuring Kerberos, refer to the MIT Kerberos documentation.

To set up the Kerberos configuration file in the default location:

1. Obtain a `krb5.conf` configuration file. You can obtain this file from your Kerberos administrator, or from the `/etc/krb5.conf` folder on the machine that is hosting the Hive Server 2 instance.
2. Rename the configuration file from `krb5.conf` to `krb5.ini`.
3. Copy the `krb5.ini` file to the `C:\ProgramData\MIT\Kerberos5` directory and overwrite the empty sample file.

To set up the Kerberos configuration file in a custom location:

1. Obtain a `krb5.conf` configuration file. You can obtain this file from your Kerberos administrator, or from the `/etc/krb5.conf` folder on the machine that is hosting the Hive Server 2 instance.
2. Place the `krb5.conf` file in an accessible directory and make note of the full path name.

3. Open the System window:

- If you are using Windows 7 or earlier, click **Start** 🪟, then right-click **Computer**, and then click **Properties**.
- Or, if you are using Windows 8 or later, right-click **This PC** on the Start screen, and then click **Properties**.

4. Click **Advanced System Settings**.

5. In the System Properties dialog box, click the **Advanced** tab and then click **Environment Variables**.

6. In the Environment Variables dialog box, under the System Variables list, click **New**.

7. In the New System Variable dialog box, in the **Variable Name** field, type **KRB5_ CONFIG**.

8. In the **Variable Value** field, type the full path to the `krb5.conf` file.

9. Click **OK** to save the new variable.

10. Make sure that the variable is listed in the System Variables list.

11. Click **OK** to close the Environment Variables dialog box, and then click **OK** to close the System Properties dialog box.

Setting Up the Kerberos Credential Cache File

Kerberos uses a credential cache to store and manage credentials.

To set up the Kerberos credential cache file:

1. Create a directory where you want to save the Kerberos credential cache file. For example, create a directory named `C:\temp`.

2. Open the System window:

- If you are using Windows 7 or earlier, click **Start** 🪟, then right-click **Computer**, and then click **Properties**.
- Or, if you are using Windows 8 or later, right-click **This PC** on the Start screen, and then click **Properties**.

3. Click **Advanced System Settings**.

4. In the System Properties dialog box, click the **Advanced** tab and then click **Environment Variables**.

5. In the Environment Variables dialog box, under the System Variables list, click **New**.

6. In the New System Variable dialog box, in the **Variable Name** field, type **KRB5CCNAME**.

7. In the **Variable Value** field, type the path to the folder you created above, and then append the file name `krb5cache`. For example, if you created the folder `C:\temp`, then type `C:\temp\krb5cache`.

> ✏ **Note:**
>
> `krb5cache` is a file (not a directory) that is managed by the Kerberos software, and it should not be created by the user. If you receive a permission error when you first use Kerberos, make sure that the `krb5cache` file does not already exist as a file or a directory.

8. Click **OK** to save the new variable.

9. Make sure that the variable appears in the System Variables list.

10. Click **OK** to close the Environment Variables dialog box, and then click **OK** to close the System Properties dialog box.

11. To make sure that Kerberos uses the new settings, restart your machine.

Obtaining a Ticket for a Kerberos Principal

A principal refers to a user or service that can authenticate to Kerberos. To authenticate to Kerberos, a principal must obtain a ticket by using a password or a keytab file. You can specify a keytab file to use, or use the default keytab file of your Kerberos configuration.

To obtain a ticket for a Kerberos principal using a password:

1. Open MIT Kerberos Ticket Manager.

2. In MIT Kerberos Ticket Manager, click **Get Ticket**.

3. In the Get Ticket dialog box, type your principal name and password, and then click **OK**.

   If the authentication succeeds, then your ticket information appears in MIT Kerberos Ticket Manager.

To obtain a ticket for a Kerberos principal using a keytab file:

1. Open a command prompt:
   - If you are using Windows 7 or earlier, click **Start** 🪟, then click **All Programs**, then click **Accessories**, and then click **Command Prompt**.
   - If you are using Windows 8 or later, click the arrow button at the bottom of the Start screen, then find the Windows System program group, and then click **Command Prompt**.

2. In the Command Prompt, type a command using the following syntax:

```
kinit -k -t [KeytabPath][Principal]
```

[KeytabPath] is the full path to the keytab file. For example:
`C:\mykeytabs\myUser.keytab`.

[Principal] is the Kerberos user principal to use for authentication. For example: `myUser@EXAMPLE.COM`.

3. If the cache location KRB5CCNAME is not set or used, then use the `-c` option of the `kinit` command to specify the location of the credential cache. In the command, the `-c` argument must appear last. For example:

```
kinit -k -t C:\mykeytabs\myUser.keytab
myUser@EXAMPLE.COM -c C:\ProgramData\MIT\krbcache
```

`Krbcache` is the Kerberos cache file, not a directory.

To obtain a ticket for a Kerberos principal using the default keytab file:

> ✎ **Note:**
>
> For information about configuring a default keytab file for your Kerberos configuration, refer to the MIT Kerberos documentation.

1. Open a command prompt:
    - If you are using Windows 7 or earlier, click **Start** 🌐, then click **All Programs**, then click **Accessories**, and then click **Command Prompt**.
    - If you are using Windows 8 or later, click the arrow button at the bottom of the Start screen, then find the Windows System program group, and then click **Command Prompt**.
2. In the Command Prompt, type a command using the following syntax:

```
kinit -k [principal]
```

[principal] is the Kerberos user principal to use for authentication. For example: `MyUser@EXAMPLE.COM`.

3. If the cache location KRB5CCNAME is not set or used, then use the `-c` option of the `kinit` command to specify the location of the credential cache. In the command, the `-c` argument must appear last. For example:

```
kinit -k -t C:\mykeytabs\myUser.keytab
myUser@EXAMPLE.COM -c C:\ProgramData\MIT\krbcache
```

`Krbcache` is the Kerberos cache file, not a directory.

Verifying the Driver Version Number on Windows

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

To verify the driver version number on Windows:

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

   > ✎ **Note:**
   >
   > Make sure to select the ODBC Data Source Administrator that has the same bitness as the client application that you are using to connect to Hive.

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

macOS Driver

macOS System Requirements

The Simba Hive ODBC Driver supports Apache Hive versions 1.0.0 through 3.1.0.

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

- macOS version 10.13 or 10.14
- 100MB of available disk space
- One of the following ODBC driver managers installed:
    - iODBC 3.52.9 or later
    - unixODBC 2.2.14 or later

Installing the Driver on macOS

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

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

To install the Simba Hive ODBC Driver on macOS:

1. Double-click **SimbaHiveODBC.dmg** to mount the disk image.
2. Double-click **SimbaHiveODBC.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/hive` directory.

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

8. If you received a license file through email, then copy the license file into the `/lib` subfolder in the driver installation directory. You must have root privileges when changing the contents of this folder.

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

Next, configure the environment variables on your machine to make sure that the ODBC driver manager can work with the driver. For more information, see Configuring the ODBC Driver Manager on Non-Windows Machines on page 45.

Verifying the Driver Version Number on macOS

If you need to verify the version of the Simba Hive 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 simba.hiveodbc
```

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

Linux Driver

For most Linux distributions, you can install the driver using the RPM file or the tarball package. If you are installing the driver on a Debian machine, you must use the Debian package.

Linux System Requirements

The Simba Hive ODBC Driver supports Apache Hive versions 1.0.0 through 3.1.0.

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

- One of the following distributions:
  - Red Hat® Enterprise Linux® (RHEL) 6 or 7
  - CentOS 6 or 7
  - SUSE Linux Enterprise Server (SLES) 11 or 12
  - Debian 8 or 9
  - Ubuntu 14.04, 16.04, or 18.04
  - Oracle Linux 7.5 or 7.6
- 150 MB of available disk space
- One of the following ODBC driver managers installed:
  - iODBC 3.52.9 or later
  - unixODBC 2.2.14 or later
- All of the following `libsasl` libraries installed:
  - `cyrus-sasl-2.1.22-7` or later
  - `cyrus-sasl-gssapi-2.1.22-7` or later
  - `cyrus-sasl-plain-2.1.22-7` or later

> ✎ **Note:**
>
> If the package manager in your Linux distribution cannot resolve the dependencies automatically when installing the driver, then download and manually install the packages.

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

Installing the Driver Using the RPM File

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

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

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

The placeholders in the file names are defined as follows:

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

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

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

1. Log in as the root user.
2. Navigate to the folder containing the RPM package for the driver.
3. Depending on the Linux distribution that you are using, run one of the following commands from the command line, where *[RPMFileName]* is the file name of the RPM package:
    - If you are using Red Hat Enterprise Linux or CentOS, run the following command:

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

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

      ```
      zypper install [RPMFileName]
      ```

    The Simba Hive ODBC Driver files are installed in the `/opt/simba/hive` directory.

> ✎ **Note:**
>
> If the package manager in your Linux distribution cannot resolve the `libsasl` dependencies automatically when installing the driver, then download and manually install the packages.

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

Next, configure the environment variables on your machine to make sure that the ODBC driver manager can work with the driver. For more information, see Configuring the ODBC Driver Manager on Non-Windows Machines on page 45.

Installing the Driver Using the Tarball Package

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

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

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

To install the driver using the tarball package:

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

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

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

The Simba Hive ODBC Driver files are installed in the `opt/simba/hive` directory.

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

Next, configure the environment variables on your machine to make sure that the ODBC driver manager can work with the driver. For more information, see Configuring the ODBC Driver Manager on Non-Windows Machines on page 45.

Installing the Driver on Debian

To install the driver on a Debian machine, use the Debian package instead of the RPM file or tarball package.

On 64-bit editions of Debian, 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:

- `simbahive_[Version]-[Release]_i386.deb` for the 32-bit driver
- `simbahive_[Version]-[Release]_amd64.deb` for the 64-bit driver

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

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

To install the Simba Hive ODBC Driver on Debian:

1. Log in as the root user, and then navigate to the folder containing the Debian package for the driver.
2. Double-click **simbahive_[Version]-[Release]_i386.deb** or **simbahive_[Version]-[Release]_amd64.deb**.
3. Follow the instructions in the installer to complete the installation process.

   The Simba Hive ODBC Driver files are installed in the `/opt/simba/hive` directory.

   > ✎ **Note:**
   >
   > If the package manager in your Ubuntu distribution cannot resolve the `libsasl` dependencies automatically when installing the driver, then download and manually install the packages required by the version of the driver that you want to install.

4. If you received a license file via email, then copy the license file into the `/opt/simba/hive/lib/32` or `/opt/simba/hiveodbc/lib/64` folder,

depending on the version of the driver that you installed. You must have root privileges when changing the contents of this folder.

Next, configure the environment variables on your machine to make sure that the ODBC driver manager can work with the driver. For more information, see Configuring the ODBC Driver Manager on Non-Windows Machines on page 45.

Verifying the Driver Version Number on Linux

If you need to verify the version of the Simba Hive ODBC Driver that is installed on your Linux machine, you can query the version number through the command-line interface if the driver was installed using an RPM file or Debian package. Alternatively, you can search the driver's binary file for version number information.

To verify the driver version number on Linux using the command-line interface:

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

-
```
yum list | grep SimbaHiveODBC
```

-
```
rpm -qa | grep SimbaHiveODBC
```

-
```
dpkg -l | grep simbahiveodbc
```

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

To verify the driver version number on Linux using the binary file:

1. Navigate to the /lib subfolder in your driver installation directory. By default, the path to this directory is: /opt/simba/hive/lib.
2. Open the driver's .so binary file in a text editor, and search for the text $driver_version_sb$:. The driver's version number is listed after this text.

AIX Driver

AIX System Requirements

The Simba Hive ODBC Driver supports Apache Hive versions 1.0.0 through 3.1.0.

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:

- IBM AIX 5.3, 6.1, or 7.1
- 150 MB of available disk space
- One of the following ODBC driver managers installed:
    - iODBC 3.52.9 or later
    - unixODBC 2.2.14 or later

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

Installing the Driver on AIX

On 64-bit editions of AIX, 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:

- `SimbaHiveODBC-32bit-[Version]-[Release].ppc.rpm` for the 32-bit driver
- `SimbaHiveODBC-[Version]-[Release].ppc.rpm` for the 64-bit driver

[Version] is the version number of the driver, and *[Release]* is the release number for this version of the driver.

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

To install the Simba Hive ODBC Driver on AIX:

1. Log in as the root user, and then navigate to the folder containing the RPM package for the driver.
2. Run the following command from the command line, where *[RPMFileName]* is the file name of the RPM package:

```
rpm --install [RPMFileName]
```

The Simba Hive ODBC Driver files are installed in the `/opt/simba/hive` directory.

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

Next, configure the environment variables on your machine to make sure that the ODBC driver manager can work with the driver. For more information, see Configuring the ODBC Driver Manager on Non-Windows Machines on page 45.

Verifying the Driver Version Number on AIX

If you need to verify the version of the Simba Hive ODBC Driver that is installed on your AIX machine, you can query the version number through the command-line interface.

To verify the driver version number on AIX:

➢ At the command prompt, run the following command:

```
rpm -qa | grep SimbaHiveODBC
```

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

Solaris Driver

Solaris System Requirements

The Simba Hive ODBC Driver supports Apache Hive versions 1.0.0 through 3.1.0.

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:

- Solaris 10 or 11 (SPARC and SPARC64 editions only)
- 150 MB of available disk space
- One of the following ODBC driver managers installed:
    - iODBC 3.52.9 or later
    - unixODBC 2.2.14 or later

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

Installing the Driver on Solaris

The Simba Hive ODBC Driver is available for Solaris as a tarball package named `Simba Hive ODBC Driver_Solaris-_gcc_[Version].[Release]_ Solaris.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 sparc64 editions of Solaris, you can execute both sparc and sparc64 applications. However, sparc64 applications must use 64-bit drivers, and sparc 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 Hive ODBC Driver on Solaris:

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

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

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

The Simba Hive ODBC Driver files are installed in the `/opt/simba/hive/` directory.

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

Next, configure the environment variables on your machine to make sure that the ODBC driver manager can work with the driver. For more information, see Configuring the ODBC Driver Manager on Non-Windows Machines on page 45.

Verifying the Driver Version Number on Solaris

If you need to verify the version of the Simba Hive ODBC Driver that is installed on your Solaris machine, you can query the version number through the command-line interface.

To verify the driver version number on Solaris:

➢ At the command prompt, run the following command:

```
rpm -qa | grep SimbaHiveODBC
```

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

Configuring the ODBC Driver Manager on Non-Windows Machines

To make sure that the ODBC driver manager on your machine is configured to work with the Simba Hive 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 45.
- 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 46.

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

Specifying ODBC Driver Managers on Non-Windows Machines

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

macOS

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

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

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

Linux or AIX

If you are using a Linux or AIX 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
or AIX shell documentation.

Solaris

If you are using a Solaris machine, then set the LD_LIBRARY_PATH environment
variable to include the paths to the ODBC driver manager libraries and the third-party
libraries that are installed with the driver. For example, if the driver manager libraries
are installed in `/usr/local/lib` and the 32-bit driver is installed in
`/opt/simba/hive`, 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:/opt/simba/hive/lib/32
```

For information about setting an environment variable permanently, refer to the Solaris
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.hiveodbc.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 SIMBAHIVEINI to the full path and file name of the `simba.hiveodbc.ini`
  file.

If you are using unixODBC, do the following:

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

- Set SIMBAHIVEINI to the full path and file name of the `simba.hiveodbc.ini` file.

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

For unixODBC:

```
export ODBCINI=/usr/local/odbc/odbc.ini
export ODBCSYSINI=/usr/local/odbc
export SIMBAHIVEINI=/etc/simba.hiveodbc.ini
```

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

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

Configuring ODBC Connections on a Non-Windows Machine

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

Creating a Data Source Name on a Non-Windows Machine

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

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

The following instructions describe how to create a DSN by specifying connection settings in the `odbc.ini` file. If your machine is already configured to use an existing `odbc.ini` file, then update that file by adding the settings described below. Otherwise, copy the `odbc.ini` file from the `Setup` subfolder in the driver installation directory to the home directory, and then update the file as described below.

For information about specifying settings in a connection string, see Configuring a DSN-less Connection on a Non-Windows Machine on page 52 and Using a Connection String on page 68. For information about driver-wide settings, see Setting Driver-Wide Configuration Options on a Non-Windows Machine on page 63.

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 Hive ODBC Driver
   ```

   As another example, for a 32-bit driver on a Linux/AIX/Solaris/Debian machine:

   ```
   [ODBC Data Sources]
   Sample DSN=Simba Hive 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/hive/lib/libhiveodbc_sbu.dylib
      ```

      As another example, for a 32-bit driver on a Linux/AIX/Solaris/Debian machine:

      ```
      Driver=/opt/simba/hive/lib/32/libhiveodbc_sb32.so
      ```

   b. Set the `HiveServerType` property to one of the following values:
      - If you are running Hive Server 1, set the property to `1`.
      - If you are running Hive Server 2, set the property to `2`.

      For example:

      ```
      HiveServerType=1
      ```

   c. Specify whether the driver uses the ZooKeeper service when connecting to Hive, and provide the necessary connection information:

- To connect to Hive without using the ZooKeeper service do the following:
    a. Set the `ServiceDiscoveryMode` property to `0`.
    b. Set the `Host` property to the IP address or host name of the server.
    c. Set the `Port` property to the number of the TCP port that the server uses to listen for client connections

    For example:

    ```
    ServiceDiscoveryMode=0
    Host=192.168.222.160
    Port=10000
    ```

- Or, to discover Hive Server 2 services through the ZooKeeper service, set properties as described in Configuring Service Discovery Mode on a Non-Windows Machine on page 54.

d. If authentication is required to access the server, then specify the authentication mechanism and your credentials. For more information, see Configuring Authentication on a Non-Windows Machine on page 55.

e. If you want to connect to the server through SSL, then enable SSL and specify the certificate information. For more information, see Configuring SSL Verification on a Non-Windows Machine on page 58.

> ✎ **Note:**
>
> If the `AuthMech` property is set to `2`, SSL is not available.

f. If you want to configure server-side properties, then set them as key-value pairs using a special syntax. For more information, see Configuring Server-Side Properties on a Non-Windows Machine on page 60.

g. Optionally, set additional key-value pairs as needed to specify other optional connection settings. For detailed information about all the configuration options supported by the Simba Hive ODBC Driver, see Driver Configuration Options on page 78.

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 46.

For example, the following is an `odbc.ini` configuration file for macOS containing a DSN that connects to a Hive Thrift Server instance directly and authenticates the connection using a user name and password:

```
[ODBC Data Sources]
Sample DSN=Simba Hive ODBC Driver
[Sample DSN]
Driver=/Library/simba/hive/lib/libhiveodbc_sbu.dylib
HiveServerType=2
ServiceDiscoveryMode=0
Host=192.168.222.160
Port=10000
UID=username
PWD=userpassword
```

As another example, the following is an `odbc.ini` configuration file for a 32-bit driver on a Linux/AIX/Solaris/Debian machine, containing a DSN that connects to a HiveThrift Server instance directly and authenticates the connection using a user name and password:

```
[ODBC Data Sources]
Sample DSN=Simba Hive ODBC Driver 32-bit
[Sample DSN]
Driver=/opt/simba/hive/lib/32/libhiveodbc_sb32.so
HiveServerType=2
ServiceDiscoveryMode=0
Host=192.168.222.160
Port=10000
UID=username
PWD=userpassword
```

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 Hive 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/hive/lib/libhiveodbc_sbu.dylib
      ```

      As another example, for a 32-bit driver on a Linux/AIX/Solaris/Debian machine:

      ```
      Driver=/opt/simba/hive/lib/32/libhiveodbc_sb32.so
      ```

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

For example:

```
Description=Simba Hive 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 46.

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

```
[ODBC Drivers]
Simba Hive ODBC Driver=Installed
[Simba Hive ODBC Driver]
Description=Simba Hive ODBC Driver
Driver=/Library/simba/hive/lib/libhiveodbc_sbu.dylib
```

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

```
[ODBC Drivers]
Simba Hive ODBC Driver 32-bit=Installed
Simba Hive ODBC Driver 64-bit=Installed
[Simba Hive ODBC Driver 32-bit]
Description=Simba Hive ODBC Driver (32-bit)
Driver=/opt/simba/hive/lib/32/libhiveodbc_sb32.so
[Simba Hive ODBC Driver 64-bit]
Description=Simba Hive ODBC Driver (64-bit)
Driver=/opt/simba/hive/lib/64/libhiveodbc_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 68.

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

For detailed information about all the connection properties that the driver supports, see Driver Configuration Options on page 78.

Configuring Service Discovery Mode on a Non-Windows Machine

You can configure the Simba Hive ODBC Driver to discover Hive Server 2 services through ZooKeeper.

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

To enable Service Discovery Mode on a non-Windows machine:

1. Set the `ServiceDiscoveryMode` connection attribute to `1`.
2. Set the `ZKNamespace` connection attribute to specify the namespace on ZooKeeper under which Hive Server 2 znodes are added.
3. Set the `Host` connection attribute to specify the ZooKeeper ensemble as a comma-separated list of ZooKeeper servers. For example, type the following, where *[ZK_Host]* is the IP address or host name of the ZooKeeper server and *[ZK_Port]* is the number of the TCP port that the ZooKeeper server uses to listen for client connections:

   ```
   [ZK_Host1]:[ZK_Port1],[ZK_Host2]:[ZK_Port2]
   ```

> **❗ Important:**
>
> When `ServiceDiscoveryMode` is set to `1`, connections to Hive Server 1 are not supported and the `Port` connection attribute is not applicable.

Configuring Authentication on a Non-Windows Machine

Some Hive Server 2 instances are configured to require authentication for access. To connect to a Hive server, you must configure the Simba Hive ODBC Driver to use the authentication mechanism that matches the access requirements of the server and provides the necessary credentials.

For information about how to determine the type of authentication your Hive server requires, see Authentication Mechanisms on page 66.

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

Depending on the authentication mechanism you use, there might be additional connection attributes that you must define. For more information about the attributes involved in configuring authentication, see Driver Configuration Options on page 78.

If cookie-based authentication is enabled in your Hive Server 2 database, you can specify a list of authentication cookies in the `HTTPAuthCookies` connection property. In this case, the driver authenticates the connection once based on the provided authentication credentials. It then uses the cookie generated by the server for each subsequent request in the same connection. For more information, see HTTPAuthCookies on page 107.

Using No Authentication

When connecting to a Hive server of type Hive Server 1, you must use No Authentication. When you use No Authentication, Binary is the only Thrift transport protocol that is supported.

To configure a connection without authentication:

1. Set the `AuthMech` connection attribute to `0`.
2. If the Hive server is configured to use SSL, then configure SSL for the connection. For more information, see Configuring SSL Verification on a Non-Windows Machine on page 58.

Using Kerberos

Kerberos must be installed and configured before you can use this authentication mechanism. For more information, refer to the MIT Kerberos Documentation:

http://web.mit.edu/kerberos/krb5-latest/doc/.

This authentication mechanism is available only for Hive Server 2 on non-HDInsight distributions. When you use Kerberos authentication, the Binary transport protocol is not supported.

To configure Kerberos authentication:

1. Set the `AuthMech` connection attribute to `1`.
2. Choose one:
    - To use the default realm defined in your Kerberos setup, do not set the `KrbRealm` attribute.
    - Or, if your Kerberos setup does not define a default realm or if the realm of your Hive server is not the default, then set the appropriate realm using the `KrbRealm` attribute.
3. Optionally, if you are using MIT Kerberos and a Kerberos realm is specified using the `KrbRealm` connection attribute, then choose one:
    - To have the Kerberos layer canonicalize the server's service principal name, leave the `ServicePrincipalCanonicalization` attribute set to `1`.
    - Or, to prevent the Kerberos layer from canonicalizing the server's service principal name, set the `ServicePrincipalCanonicalization` attribute to `0`.
4. Set the `KrbHostFQDN` attribute to the fully qualified domain name of the Hive Server 2 host.

> ✎ **Note:**
>
> To use the Hive server host name as the fully qualified domain name for Kerberos authentication, set `KrbHostFQDN` to `_HOST`.

5. Set the `KrbServiceName` attribute to the service name of the Hive .
6. To allow the driver to pass your credentials directly to the server for use in authentication, set `DelegateKrbCreds` to `1`.
7. Set the `ThriftTransport` connection attribute to the transport protocol to use in the Thrift layer.

> ❗ **Important:**
>
> When using this authentication mechanism, Binary (`ThriftTransport=0`) is not supported.

8. If the Hive server is configured to use SSL, then configure SSL for the connection. For more information, see Configuring SSL Verification on a Non-Windows Machine on page 58.

Using User Name

This authentication mechanism requires a user name but does not require a password. The user name labels the session, facilitating database tracking.

This authentication mechanism is available only for Hive Server 2 on non-HDInsight distributions. Most default configurations of Hive Server 2 require User Name authentication. When you use User Name authentication, SSL is not supported and SASL is the only Thrift transport protocol available.

To configure User Name authentication:

1. Set the `AuthMech` connection attribute to `2`.
2. Set the `UID` attribute to an appropriate user name for accessing the Hive server.

Using User Name And Password

This authentication mechanism requires a user name and a password.

This authentication mechanism is available only for Hive Server 2 on non-HDInsight distributions.

To configure User Name And Password authentication:

1. Set the `AuthMech` connection attribute to `3`.
2. Set the `UID` attribute to an appropriate user name for accessing the Hive server.
3. Set the `PWD` attribute to the password corresponding to the user name you provided above.
4. Set the `ThriftTransport` connection attribute to the transport protocol to use in the Thrift layer.
5. If the Hive server is configured to use SSL, then configure SSL for the connection. For more information, see Configuring SSL Verification on a Non-Windows Machine on page 58.

Using Windows Azure HDInsight Service

This authentication mechanism is available only for Hive Server 2 on HDInsight distributions. When you use this authentication mechanism, you must enable SSL, and HTTP is the only Thrift transport protocol available.

To configure a connection to a Hive server on Windows Azure HDInsight Service:

1. Set the `AuthMech` connection attribute to `6`.
2. Set the `HTTPPath` attribute to the partial URL corresponding to the Hive server.

3. Set the `UID` attribute to an appropriate user name for accessing the Hive server.

4. Set the `PWD` attribute to the password corresponding to the user name you typed above.

5. If necessary, you can create custom HTTP headers. For more information, see http.header. on page 107.

6. Configure SSL settings as needed. For more information, see Configuring SSL Verification on a Non-Windows Machine on page 58.

Delegating Authentication to a Specific User

Some Hive Server 2 instances support the ability to delegate all operations against Hive to the specified user, rather than to the authenticated user for the connection.

To delegate all operations to a specified user:

➢ Set the `DelegationUID` configuration option to the name of the user to be delegated.

If the server returns an error message such as **Failed to validate proxy privilege of [RealUser] for [DelegationUID]**, you may need to modify the server's `core-site.xml` configuration file, as follows:

1. In the server's `core-site.xml` configuration file, add the following properties:

```
hadoop.proxyuser.[RealUser].groups=*
hadoop.proxyuser.[RealUser].hosts=*
```

Where *[RealUser]* is the authenticated user for the connection.

2. If you are using Kerberos authentication, then in the server's `core-site.xml` configuration file, add the following properties:

```
hadoop.proxyuser.[Principal].groups=*
hadoop.proxyuser.[Principal].hosts=*
```

Where *[Principal]* is the primary Kerberos principal user. For example, if the primary Kerberos principal user is `kerbuser@example.com`, replace *[Principal]* with `kerbuser`.

For more information on resolving this error, see your Hive Server documentation.

Configuring SSL Verification on a Non-Windows Machine

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

connect to a server, the driver supports identity verification between the client (the driver itself) and the server.

✏ **Note:**

If the `AuthMech` property is set to `2`, SSL is not available.

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

To configure SSL verification on a non-Windows machine:

1. To enable SSL connections, set the `SSL` attribute to `1`.
2. To allow authentication using self-signed certificates that have not been added to the list of trusted certificates, set the `AllowSelfSignedServerCert` attribute to `1`.
3. To allow the common name of a CA-issued SSL certificate to not match the host name of the Hive server, set the `CAIssuedCertNamesMismatch` attribute to `1`.
4. Choose one:
   - To configure the driver to load SSL certificates from a specific `.pem` file when verifying the server, set the `TrustedCerts` attribute to the full path of the `.pem` file.
   - Or, to use the trusted CA certificates `.pem` file that is installed with the driver, do not specify a value for the `TrustedCerts` attribute.
5. To configure two-way SSL verification, set the `TwoWaySSL` attribute to `1` and then do the following:
   a. Set the `ClientCert` attribute to the full path of the `.pem` file containing the client's certificate.
   b. Set the `ClientPrivateKey` attribute to the full path of the file containing the client's private key.
   c. If the private key file is protected with a password, set the `ClientPrivateKeyPassword` attribute to the password.
6. To specify the minimum version of TLS to use, set the `Min_TLS` property to the minimum version of TLS. Supported options include `1.0` for TLS 1.0, `1.1` for TLS 1.1, and `1.2` for TLS 1.2.

Configuring Server-Side Properties on a Non-Windows Machine

You can use the driver to apply configuration properties to the Hive server.

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

To configure server-side properties on a non-Windows machine:

1. To set a server-side property, use the syntax `SSP_[SSPKey]=[SSPValue]`, where *[SSPKey]* is the name of the server-side property and *[SSPValue]* is the value to specify for that property. For example, to set the `mapreduce.job.queuename` property to `myQueue`, type the following:

   ```
   SSP_mapreduce.job.queuename=myQueue
   ```

   > ✎ **Note:**
   >
   > - When setting a server-side property in a connection string, it is recommended that you enclose the value in braces (`{ }`) to make sure that special characters can be properly escaped.
   > - For a list of all Hadoop and Hive server-side properties that your implementation supports, type `set -v` at the Hive CLI command line or Beeline. You can also execute the `set -v` query after connecting using the driver.

2. To change the method that the driver uses to apply server-side properties, do one of the following:
   - To configure the driver to apply each server-side property by executing a query when opening a session to the Hive server, set the `ApplySSPWithQueries` property to `1`.
   - Or, to configure the driver to use a more efficient method for applying server-side properties that does not involve additional network round-tripping, set the `ApplySSPWithQueries` property to `0`.

   > ✎ **Note:**
   >
   > The more efficient method is not available for Hive Server 1, and it might not be compatible with some Hive Server builds. If the server-side properties do not take effect when the `ApplySSPWithQueries` property is set to `0`, then set it to `1`.

3. To disable the driver's default behavior of converting server-side property key names to all lower-case characters, set the `LCaseSspKeyName` property to `0`.

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

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

To enable logging on a non-Windows machine:

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

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

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

4. Set the `LogFileCount` key to the maximum number of log files to keep.

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

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

> ✎ **Note:**
>
> After the maximum file size is reached, the driver creates a new file and continues logging.

6. Optionally, to prefix the log file name with the user name and process ID associated with the connection, set the `UseLogPrefix` property to `1`.

7. Save the `simba.hiveodbc.ini` configuration file.

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

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

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

If you set the `UseLogPrefix` property to `1`, then each file name is prefixed with `[UserName]_[ProcessID]_`, where *[UserName]* is the user name associated with the connection and *[ProcessID]* is the process ID of the application through which the connection is made. For more information, see UseLogPrefix on page 108.

To disable logging on a non-Windows machine:

1. Open the `simba.hiveodbc.ini` configuration file in a text editor.

2. Set the `LogLevel` key to `0`.

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

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

Setting Driver-Wide Configuration Options on a Non-Windows Machine

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

> ✎ **Note:**
>
> Settings in the connection string take precedence over settings in the DSN, and settings in the DSN take precedence over driver-wide settings.

To set driver-wide configuration options on a non-Windows machine:

1. In a text editor, open the `simba.hiveodbc.ini` configuration file.
2. In the `[Driver]` section, specify configuration options as key-value pairs. Start a new line for each key-value pair.

   For example, to enable User Name authentication using "simba" as the user name, type the following:

   ```
   AuthMech=2
   UID=simba
   ```

   For detailed information about all the configuration options supported by the driver, see Driver Configuration Options on page 78.
3. Save the `simba.hiveodbc.ini` configuration file.

Testing the Connection on a Non-Windows Machine

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

Using the iODBC Driver Manager

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

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

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

To test your connection using the iODBC driver manager:

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

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.

Authentication Mechanisms

To connect to a Hive server, you must configure the Simba Hive ODBC Driver to use the authentication mechanism that matches the access requirements of the server and provides the necessary credentials. To determine the authentication settings that your Hive server requires, check the server configuration and then refer to the corresponding section below.

Hive Server 1

You must use No Authentication as the authentication mechanism. Hive Server 1 instances do not support authentication.

Hive Server 2 on an HDInsight Distribution

If you are connecting to HDInsight Service running on Windows Azure, then you must use the Windows Azure HDInsight Service mechanism.

Hive Server 2 on a non-HDInsight Distribution

> ✎ **Note:**
>
> Most default configurations of Hive Server 2 on non-HDInsight distributions require User Name authentication.

Configuring authentication for a connection to a Hive Server 2 instance on a non-HDInsight Distribution involves setting the authentication mechanism, the Thrift transport protocol, and SSL support. To determine the settings that you need to use, check the following three properties in the `hive-site.xml` file in the Hive server that you are connecting to:

- `hive.server2.authentication`
- `hive.server2.transport.mode`
- `hive.server2.use.SSL`

Use the following table to determine the authentication mechanism that you need to configure, based on the `hive.server2.authentication` value in the `hive-site.xml` file:

hive.server2.authentication	Authentication Mechanism
NOSASL	No Authentication
KERBEROS	Kerberos
NONE	User Name
LDAP	User Name and Password

Use the following table to determine the Thrift transport protocol that you need to configure, based on the `hive.server2.authentication` and `hive.server2.transport.mode` values in the `hive-site.xml` file:

hive.server2.authentication	hive.server2.transport.mode	Thrift Transport Protocol
NOSASL	binary	Binary
KERBEROS	binary or http	SASL or HTTP
NONE	binary or http	SASL or HTTP
LDAP	binary or http	SASL or HTTP

To determine whether SSL should be enabled or disabled for your connection, check the `hive.server2.use.SSL` value in the `hive-site.xml` file. If the value is true, then you must enable and configure SSL in your connection. If the value is false, then you must disable SSL in your connection.

For detailed instructions on how to configure authentication when using the Windows driver, see .

For detailed instructions on how to configure authentication when using a non-Windows driver, see .

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 78.

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:

- *[DomainName]* is the fully qualified domain name of the Hive server host.
- *[Namespace]* is the namespace on ZooKeeper under which Hive Server 2 znodes are added.
- *[PortNumber]* is the number of the TCP port that the Hive server uses to listen for client connections.
- *[Realm]* is the Kerberos realm of the Hive server host.
- *[Server]* is the IP address or host name of the Hive server to which you are connecting.
- *[ServerURL]* is the partial URL corresponding to the Hive server.

- *[ServiceName]* is the Kerberos service principal name of the Hive server.
- *[YourPassword]* is the password corresponding to your user name.
- *[YourUserName]* is the user name that you use to access the Hive server.

Connecting to a Hive Server 1 Instance

The following is the format of a DSN-less connection string that connects to a Hive Server 1 instance:

```
Driver=Simba Hive ODBC Driver;HiveServerType=1;
Host=[Server];Port=[PortNumber];
```

For example:

```
Driver=Simba Hive ODBC Driver;HiveServerType=1;
Host=192.168.222.160;Port=10000;
```

Connecting to a Standard Hive Server 2 Instance

The following is the format of a DSN-less connection string for a standard connection to a Hive Server 2 instance. By default, the driver is configured to connect to a Hive Server 2 instance that requires User Name authentication, and the driver uses **anonymous** as the user name.

```
Driver=Simba Hive ODBC Driver;Host=[Server];
Port=[PortNumber];
```

For example:

```
Driver=Simba Hive ODBC Driver;Host=192.168.222.160;
Port=10000;
```

Connecting using Dynamic Service Discovery

The following is the format of a DSN-less connection string that discovers Hive Server 2 services via the ZooKeeper service.

```
Driver=Simba Hive ODBC Driver;ServiceDiscoveryMode=1;
Host=[Server1]:[PortNumber1], [Server2]:[PortNumber2],
[Server3]:[PortNumber3];ZKNamespace=[Namespace];
```

For example:

```
Driver=Simba Hive ODBC Driver;ServiceDiscoveryMode=1;
Host=192.168.222.160:10000, 192.168.222.165:10000,
```

```
192.168.222.231:10000;ZKNamespace=hiveserver;
```

Connecting to a Hive Server 2 Instance Without Authentication

The following is the format of a DSN-less connection string for a Hive Server 2 instance that does not require authentication.

```
Driver=Simba Hive ODBC Driver;Host=[Server];
Port=[PortNumber];AuthMech=0;
```

For example:

```
Driver=Simba Hive ODBC Driver;Host=192.168.222.160;
Port=10000;AuthMech=0;
```

Connecting to a Hive Server that Requires Kerberos Authentication

The following is the format of a DSN-less connection string that connects to a Hive Server 2 instance requiring Kerberos authentication:

```
Driver=Simba Hive ODBC Driver;Host=[Server];
Port=[PortNumber];AuthMech=1;KrbRealm=[Realm];
KrbHostFQDN=[DomainName];KrbServiceName=[ServiceName];
```

For example:

```
Driver=Simba Hive ODBC Driver;Host=192.168.222.160;
Port=10000;AuthMech=1;KrbRealm=SIMBA;
KrbHostFQDN=localhost.localdomain;KrbServiceName=hive;
```

Connecting to a Hive Server that Requires User Name And Password Authentication (LDAP)

The following is the format of a DSN-less connection string that connects to a Hive Server 2 instance requiring User Name And Password authentication:

```
Driver=Simba Hive ODBC Driver;Host=[Server];
Port=[PortNumber];AuthMech=3;UID=[YourUserName];
PWD=[YourPassword];
```

For example:

```
Driver=Simba Hive ODBC Driver;Host=192.168.222.160;
Port=10000;AuthMech=3;UID=simba;PWD=simba;
```

Connecting to a Hive Server on Windows Azure HDInsight Service

The following is the format of a DSN-less connection string that connects to a Hive Server 2 instance running on Windows Azure HDInsight Service:

```
Driver=Simba Hive ODBC Driver;Host=[Server];
Port=[PortNumber];AuthMech=6;SSL=1;UID=[YourUserName];
PWD=[YourPassword];HTTPPath=[ServerURL];
```

For example:

```
Driver=Simba Hive ODBC Driver;Host=192.168.222.160;
Port=10000;AuthMech=6;SSL=1;UID=simba;PWD=simba;
HTTPPath=gateway/sandbox/hive;
```

Features

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

- SQL Connector for HiveQL on page 72
- Data Types on page 73
- Catalog and Schema Support on page 74
- hive_system Table on page 74
- Server-Side Properties on page 75
- Get Tables With Query on page 75
- Active Directory on page 75
- Write-back on page 76
- Timestamp Function Support on page 76
- Dynamic Service Discovery using ZooKeeper on page 76
- Security and Authentication on page 77

SQL Connector for HiveQL

The native query language supported by Hive is HiveQL. For simple queries, HiveQL is a subset of SQL-92. However, the syntax is different enough that most applications do not work with native HiveQL.

To bridge the difference between SQL and HiveQL, the SQL Connector feature translates standard SQL-92 queries into equivalent HiveQL queries. The SQL Connector performs syntactical translations and structural transformations. For example:

- **Quoted Identifiers**: The double quotes (**"**) that SQL uses to quote identifiers are translated into back quotes (**`**) to match HiveQL syntax. The SQL Connector needs to handle this translation because even when a driver reports the back quote as the quote character, some applications still generate double-quoted identifiers.
- **Table Aliases**: Support is provided for the AS keyword between a table reference and its alias, which HiveQL normally does not support.
- **JOIN**, **INNER JOIN**, and **CROSS JOIN**: SQL JOIN, INNER JOIN, and CROSS JOIN syntax is translated to HiveQL JOIN syntax.
- **TOP N/LIMIT**: SQL TOP N queries are transformed to HiveQL LIMIT queries.

Data Types

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

The following table lists the supported data type mappings.

Hive Type	SQL Type
BIGINT	SQL_BIGINT
BINARY	SQL_VARBINARY
BOOLEAN	SQL_BIT
CHAR(n)	SQL_CHAR ✎ **Note:** SQL_WCHAR is returned instead if the Unicode SQL Character Types configuration option (the `UseUnicodeSqlCharacterTypes` key) is enabled.
DATE	SQL_TYPE_DATE
DECIMAL	SQL_DECIMAL
DECIMAL(p,s)	SQL_DECIMAL
DOUBLE	SQL_DOUBLE
FLOAT	SQL_REAL
INT	SQL_INTEGER
SMALLINT	SQL_SMALLINT

Hive Type	SQL Type
STRING	SQL_VARCHAR ✎ **Note:** SQL_WVARCHAR is returned instead if the Unicode SQL Character Types configuration option (the `UseUnicodeSqlCharacterTypes` key) is enabled.
TIMESTAMP	SQL_TYPE_TIMESTAMP
TINYINT	SQL_TINYINT
VARCHAR(n)	SQL_VARCHAR

✎ **Note:**

The aggregate types (ARRAY, MAP, and STRUCT) are not supported. Columns of aggregate types are treated as STRING columns.

The interval types (YEAR TO MONTH and DAY TIME) are supported only in query expressions and predicates. Interval types are not supported as column data types in tables.

Catalog and Schema Support

The Simba Hive ODBC Driver supports both catalogs and schemas to make it easy for the driver to work with various ODBC applications. Since Hive only organizes tables into schemas/databases, the driver provides a synthetic catalog named HIVE under which all of the schemas/databases are organized. The driver also maps the ODBC schema to the Hive schema/database.

hive_system Table

A pseudo-table called hive_system can be used to query for Hive cluster system environment information. The pseudo-table is under the pseudo-schema called hive_system. The table has two STRING type columns, envkey and envvalue. Standard SQL can be executed against the hive_system table. For example:

```
SELECT * FROM HIVE.hive_system.hive_system WHERE envkey LIKE
'%hive%'
```

The above query returns all of the Hive system environment entries whose key contains the word "hive". A special query, `set -v`, is executed to fetch system environment information. Some versions of Hive do not support this query. For versions of Hive that do not support querying system environment information, the driver returns an empty result set.

Server-Side Properties

The Simba Hive ODBC Driver allows you to set server-side properties via a DSN. Server-side properties specified in a DSN affect only the connection that is established using the DSN.

You can also specify server-side properties for connections that do not use a DSN. To do this, use the Simba Hive ODBC Driver Configuration tool that is installed with the Windows version of the driver, or set the appropriate configuration options in your connection string or the `simba.hiveodbc.ini` file. Properties specified in the driver configuration tool or the `simba.hiveodbc.ini` file apply to all connections that use the Simba Hive ODBC Driver.

For more information about setting server-side properties when using the Windows driver, see Configuring Server-Side Properties on Windows on page 25. For information about setting server-side properties when using the driver on a non-Windows platform, see Configuring Server-Side Properties on a Non-Windows Machine on page 60.

Get Tables With Query

The Get Tables With Query configuration option allows you to choose whether to use the SHOW TABLES query or the GetTables API call to retrieve table names from a database.

Active Directory

The Simba Hive ODBC Driver supports Active Directory Kerberos on Windows. There are two prerequisites for using Active Directory Kerberos on Windows:

- MIT Kerberos is not installed on the client Windows machine.
- The MIT Kerberos Hadoop realm has been configured to trust the Active Directory realm so that users in the Active Directory realm can access services in the MIT Kerberos Hadoop realm.

Write-back

The Simba Hive ODBC Driver supports translation for the following syntax when connecting to a Hive Server 2 instance:

- INSERT
- UPDATE
- DELETE
- CREATE
- DROP

If the statement contains non-standard SQL-92 syntax, then the driver is unable to translate the statement to SQL and instead falls back to using HiveQL.

Timestamp Function Support

The Simba Hive ODBC Driver supports the following ODBC functions for working with data of type TIMESTAMP:

- **TIMESTAMPADD**: You can call this function to increment a TIMESTAMP value by a specified interval of time.
- **TIMESTAMPDIFF**: You can call this function to calculate the interval of time between two specified TIMESTAMP values.

The types of time intervals that are supported for these functions might vary depending on the Hive server version that you are connecting to. To return a list of the intervals supported for TIMESTAMPADD, call the SQLGetInfo catalog function using SQL_TIMEDATE_ADD_INTERVALS as the argument. Similarly, to return a list of the intervals supported for TIMESTAMPDIFF, call SQLGetInfo using SQL_TIMEDATE_DIFF_INTERVALS as the argument.

> ✎ **Note:**
>
> The SQL_TSI_FRAC_SECOND interval is not supported by Hive.

Dynamic Service Discovery using ZooKeeper

The Simba Hive ODBC Driver can be configured to discover Hive Server 2 services via the ZooKeeper service.

For information about configuring this feature in the Windows driver, see Creating a Data Source Name on Windows on page 9 or Configuring a DSN-less Connection on Windows on page 12. For information about configuring this feature when using the

driver on a non-Windows platform, see Configuring Service Discovery Mode on a Non-Windows Machine on page 54.

Security and Authentication

To protect data from unauthorized access, some Hive data stores require connections to be authenticated with user credentials or encrypted using the SSL protocol. The Simba Hive 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 1.0, 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 mechanisms that enable you to authenticate your connection using the Kerberos protocol, your Hive user name only, or your Hive user name and password. You can also authenticate a connection to an HDInsight distribution on Windows Azure. You must use the authentication mechanism that matches the security requirements of the Hive server. For information about determining the appropriate authentication mechanism to use based on the Hive server configuration, see Authentication Mechanisms on page 66. For detailed driver configuration instructions, see Configuring Authentication on Windows on page 15 or Configuring Authentication on a Non-Windows Machine on page 55.

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

- No identity verification
- One-way authentication
- Two-way authentication

It is recommended that you enable SSL whenever you connect to a server that is configured to support it. SSL encryption protects data and credentials when they are transferred over the network, and provides stronger security than authentication alone. For detailed configuration instructions, see Configuring SSL Verification on Windows on page 23 or Configuring SSL Verification on a Non-Windows Machine on page 58.

Driver Configuration Options

Driver Configuration Options lists the configuration options available in the Simba Hive 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 are available in the Simba Hive ODBC Driver Configuration tool and the following dialog boxes:

- Simba Hive ODBC Driver DSN Setup
- Advanced Options
- Server Side Properties
- SSL Options
- HTTP Properties

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

> ✎ **Note:**
>
> If you are using the driver on a non-Windows machine, you can set driver configuration properties in a connection string, in a DSN (in the `odbc.ini` file), or as a driver-wide setting (in the `simba.hiveodbc.ini` file). Settings in the connection string take precedence over settings in the DSN, and settings in the DSN take precedence over driver-wide settings.

Configuration Options Appearing in the User Interface

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

- Allow Common Name Host Name Mismatch on page 80
- Allow Self-Signed Server Certificate on page 80
- Apply Properties with Queries on

- Invalid Session Auto Recover on page 91
- Log Level on page 91
- Log Path on page 92
- Max File Size on page 93

Allow Common Name Host Name Mismatch

Key Name	Default Value	Required
`CAIssuedCertNamesMismatch`	Clear (0)	No

Description

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

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

> ✎ **Note:**
>
> This setting is applicable only when SSL is enabled.

Allow Self-Signed Server Certificate

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

Description

This option specifies whether the driver allows a connection to a Hive server that uses a self-signed certificate, even if this certificate is not in the list of trusted certificates. This list is contained in the Trusted Certificates file, or in the system Trust Store if the system Trust Store is used instead of a file.

- Enabled (1): The driver authenticates the Hive server even if the server is using a self-signed certificate that has not been added to the list of trusted certificates.
- Disabled (0): The driver does not allow self-signed certificates from the server unless they have already been added to the list of trusted certificates.

> ✎ **Note:**
>
> This setting is applicable only when SSL is enabled.

Apply Properties with Queries

Key Name	Default Value	Required
ApplySSPWithQueries	Selected (1)	No

Description

This option specifies how the driver applies server-side properties.

- Enabled (1): The driver applies each server-side property by executing a set *SSPKey=SSPValue* query when opening a session to the Hive server.
- Disabled (0): The driver uses a more efficient method for applying server-side properties that does not involve additional network round-tripping. However, some Hive Server 2 builds are not compatible with the more efficient method.

> ✎ **Note:**
>
> When connecting to a Hive Server 1 instance, this option is always enabled.

Async Exec Poll Interval

Key Name	Default Value	Required
AsyncExecPollInterval	100	No

Description

The time in milliseconds between each poll for the query execution status.

"Asynchronous execution" refers to the fact that the RPC call used to execute a query against Hive is asynchronous. It does not mean that ODBC asynchronous operations are supported.

> ✎ **Note:**
>
> This option is applicable only to HDInsight clusters.

Binary Column Length

Key Name	Default Value	Required
BinaryColumnLength	32767	No

Description

The maximum data length for BINARY columns.

By default, the columns metadata for Hive does not specify a maximum data length for BINARY columns.

Canonicalize Principal FQDN

Key Name	Default Value	Required
ServicePrincipal Canonicalization	Selected (1)	No

Description

This option specifies whether the Kerberos layer canonicalizes the host FQDN in the server's service principal name.

- Enabled (1): The Kerberos layer canonicalizes the host FQDN in the server's service principal name.
- Disabled (0): The Kerberos layer does not canonicalize the host FQDN in the server's service principal name.

> ✎ **Note:**
>
> - This option only affects MIT Kerberos, and is ignored when using Active Directory Kerberos.
> - This option can only be disabled if the Kerberos Realm or KrbRealm key is specified.

Check Certificate Revocation

Key Name	Default Value	Required
CheckCertRevocation	Selected (1)	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 property is disabled when the `AllowSelfSignedServerCert` property is set to `1`.

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

Client Certificate File

Key Name	Default Value	Required
ClientCert	None	No

Description

The full path to the `.pem` file containing the client's SSL certificate.

> ✎ **Note:**
>
> This setting is applicable only when two-way SSL is enabled.

Client Private Key File

Key Name	Default Value	Required
ClientPrivateKey	None	Yes, if two-way SSL verification is enabled.

Description

The full path to the `.pem` file containing the client's SSL private key.

If the private key file is protected with a password, then provide the password using the driver configuration option .

> ✎ **Note:**
>
> This setting is applicable only when two-way SSL is enabled.

Client Private Key Password

Key Name	Default Value	Required
ClientPrivateKeyPassword	None	Yes, if two-way SSL verification is enabled and the client's private key file is protected with a password.

Description

The password of the private key file that is specified in the Client Private Key File field (`ClientPrivateKey`).

Convert Key Name to Lower Case

Key Name	Default Value	Required
LCaseSspKeyName	Selected (`1`)	No

Description

This option specifies whether the driver converts server-side property key names to all lower-case characters.

- Enabled (`1`): The driver converts server-side property key names to all lower-case characters.
- Disabled (`0`): The driver does not modify the server-side property key names.

Database

Key Name	Default Value	Required
Schema	default	No

Description

The name of the database schema to use when a schema is not explicitly specified in a query. You can still issue queries on other schemas by explicitly specifying the schema in the query.

> ✏️ **Note:**
>
> To inspect your databases and determine the appropriate schema to use, at the Hive command prompt, type `show databases`.

Decimal Column Scale

Key Name	Default Value	Required
DecimalColumnScale	10	No

Description

The maximum number of digits to the right of the decimal point for numeric data types.

Default String Column Length

Key Name	Default Value	Required
Default StringColumnLength	255	No

Description

The maximum number of characters that can be contained in STRING columns.

By default, the columns metadata for Hive does not specify a maximum length for STRING columns.

Delegate Kerberos Credentials

Key Name	Default Value	Required
DelegateKrbCreds	Clear (0)	No

Description

This option specifies whether your Kerberos credentials are forwarded to the server and used for authentication.

> ✎ **Note:**
>
> This option is only applicable when Authentication Mechanism is set to Kerberos (AuthMech=1).

Delegation UID

Key Name	Default Value	Required
DelegationUID	None	No

Description

If a value is specified for this setting, the driver delegates all operations against Hive to the specified user, rather than to the authenticated user for the connection.

> ✎ **Note:**
>
> This option is applicable only when connecting to a Hive Server 2 instance that supports this feature.

Driver Config Take Precedence

Key Name	Default Value	Required
DriverConfigTakePrecedence	Clear (0)	No

Description

This option specifies whether driver-wide configuration settings take precedence over connection and DSN settings.

- Enabled (1): Driver-wide configurations take precedence over connection and DSN settings.
- Disabled (0): Connection and DSN settings take precedence instead.

Enable Auto Reconnect

Key Name	Default Value	Required
AutoReconnect	Selected (1)	Yes

Description

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

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

Enable SSL

Key Name	Default Value	Required
SSL	Clear (0)	No

Description

This option specifies whether the client uses an SSL encrypted connection to communicate with the Hive server.

- Enabled (1): The client communicates with the Hive server using SSL.
- Disabled (0): SSL is disabled.

SSL is configured independently of authentication. When authentication and SSL are both enabled, the driver performs the specified authentication method over an SSL connection.

✎ Note:

- This option is applicable only when connecting to a Hive server that supports SSL.
- If you selected User Name as the authentication mechanism, SSL is not available.

Fast SQLPrepare

Key Name	Default Value	Required
FastSQLPrepare	Clear (0)	No

Description

This option specifies whether the driver defers query execution to SQLExecute.

- Enabled (1): The driver defers query execution to SQLExecute.
- Disabled (0): The driver does not defer query execution to SQLExecute.

✎ Note:

When using Native Query mode, the driver executes the HiveQL query to retrieve the result set metadata for SQLPrepare. As a result, SQLPrepare might be slow. If the result set metadata is not required after calling SQLPrepare, then enable Fast SQLPrepare.

Get Tables With Query

Key Name	Default Value	Required
GetTablesWithQuery	0	No

Description

This option specifies whether the driver uses the SHOW TABLES query or the GetTables Thrift API call to retrieve table names from the database.

- Enabled (1): The driver uses the SHOW TABLES query to retrieve table names.
- Disabled (0): The driver uses the GetTables Thrift API call to retrieve table names.

> ✏ **Note:**
>
> This option is applicable only when connecting to a Hive Server 2 instance.

Hive Server Type

Key Name	Default Value	Required
HiveServerType	Hive Server 2 (2)	No

Description

This option specifies the type of Hive server.

- Hive Server 1 (1): The driver connects to a Hive Server 1 instance.
- Hive Server 2 (2): The driver connects to a Hive Server 2 instance.

> ✏ **Note:**
>
> If Service Discovery Mode is enabled, then connections to Hive Server 1 are not supported.

Host(s)

Key Name	Default Value	Required
Host	None	Yes

Description

If Service Discovery Mode is disabled, the IP address or host name of the Hive server.

If Service Discovery Mode is enabled, specify a comma-separated list of ZooKeeper servers in the following format, where *[ZK_Host]* is the IP address or host name of the ZooKeeper server and *[ZK_Port]* is the number of the TCP port that the ZooKeeper server uses to listen for client connections:

```
[ZK_Host1]:[ZK_Port1],[ZK_Host2]:[ZK_Port2]
```

Host FQDN

Key Name	Default Value	Required
KrbHostFQDN	_HOST	No

Description

The fully qualified domain name of the Hive Server 2 host.

When the value of Host FQDN is `_HOST`, the driver uses the Hive server host name as the fully qualified domain name for Kerberos authentication. If Service Discovery Mode is disabled, then the driver uses the value specified in the Host connection attribute. If Service Discovery Mode is enabled, then the driver uses the Hive Server 2 host name returned by ZooKeeper.

HTTP Path

Key Name	Default Value	Required
HTTPPath	`/hive2` if using Windows Azure HDInsight Service (`6`). `/` if using non-Windows Azure HDInsight Service with Thrift Transport set to HTTP (`2`).	No

Description

The partial URL corresponding to the Hive server.

The driver forms the HTTP address to connect to by appending the HTTP Path value to the host and port specified in the DSN or connection string. For example, to connect to the HTTP address `http://localhost:10002/gateway/sandbox/hive/version`, you would set HTTP Path to `/gateway/sandbox/hive/version`.

> ✎ **Note:**
>
> By default, Hive servers use `cliservice` as the partial URL.

Invalid Session Auto Recover

Key Name	Default Value	Required
InvalidSessionAutoRecover	Selected (1)	No

Description

This option specifies whether the driver automatically opens a new session when the existing session is no longer valid.

- Enabled (1): The driver automatically opens a new session when the existing session is no longer valid.
- Disabled (0): The driver does not automatically open new sessions.

> ✎ **Note:**
>
> This option is applicable only when connecting to Hive Server 2.

Log Level

Key Name	Default Value	Required
LogLevel	OFF (0)	No

Description

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

> ❗ **Important:**
>
> - Only enable logging long enough to capture an issue. Logging decreases performance and can consume a large quantity of disk space.
> - The settings for logging apply to every connection that uses the Simba Hive ODBC Driver, so make sure to disable the feature after you are done using it.
> - This option is not supported in connection strings. To configure logging for the Windows driver, you must use the Logging Options dialog box. To configure logging for a non-Windows driver, you must use the simba.hiveodbc.ini file.

Set the property to one of the following values:

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

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

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

If you enable the `UseLogPrefix` connection property, the driver prefixes the log file name with the user name associated with the connection and the process ID of the application through which the connection is made. For more information, see

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.hiveodbc.ini` file.

Max File Size

Key Name	Default Value	Required
LogFileSize	20971520	No

Description

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

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

> ❗ **Important:**
>
> 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.hiveodbc.ini` file.

Max Number Files

Key Name	Default Value	Required
LogFileCount	50	No

Description

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

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

Mechanism

Key Name	Default Value	Required
AuthMech	No Authentication (0) if you are connecting to Hive Server 1. User Name (2) if you are connecting to Hive Server 2.	No

Description

The authentication mechanism to use.

Select one of the following settings, or set the key to the corresponding number:

- No Authentication (0)
- Kerberos (1)
- User Name (2)
- User Name And Password (3)
- Windows Azure HDInsight Service (6)

Minimum TLS

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

Description

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

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

Password

Key Name	Default Value	Required
PWD	None	Yes, if the authentication mechanism is User Name And Password (3) or Windows Azure HDInsight Service (6).

Description

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

Port

Key Name	Default Value	Required
Port	• non-HDInsight clusters: 10000 • Windows Azure HDInsight Service: 443	Yes, if Service Discovery Mode is disabled.

Description

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

Realm

Key Name	Default Value	Required
KrbRealm	Depends on your Kerberos configuration.	No

Description

The realm of the Hive Server 2 host.

If your Kerberos configuration already defines the realm of the Hive Server 2 host as the default realm, then you do not need to configure this option.

Rows Fetched Per Block

Key Name	Default Value	Required
RowsFetchedPerBlock	10000	No

Description

The maximum number of rows that a query returns at a time.

Valid values for this setting include any positive 32-bit integer. However, testing has shown that performance gains are marginal beyond the default value of 10000 rows.

Save Password (Encrypted)

Key Name	Default Value	Required
N/A	Selected	No

Description

This option specifies whether the password is saved in the registry.

- Enabled: The password is saved in the registry.
- Disabled: The password is not saved in the registry.

This option is available only in the Windows driver. It appears in the Simba Hive ODBC Driver DSN Setup dialog box and the SSL Options dialog box.

> ❗ **Important:**
>
> The password is obscured (not saved in plain text). However, it is still possible for the encrypted password to be copied and used.

Service Discovery Mode

Key Name	Default Value	Required
ServiceDiscoveryMode	No Service Discovery (0)	No

Description

This option specifies whether the driver uses the ZooKeeper service.

- Enabled (1): The driver discovers Hive Server 2 services via the ZooKeeper service.
- Disabled (0): The driver connects to Hive without using a discovery service.

Service Name

Key Name	Default Value	Required
KrbServiceName	None	No

Description

The Kerberos service principal name of the Hive server.

Show System Table

Key Name	Default Value	Required
ShowSystemTable	Clear (0)	No

Description

This option specifies whether the driver returns the hive_system table for catalog function calls such as SQLTables and SQLColumns.

- Enabled (1): The driver returns the hive_system table for catalog function calls such as SQLTables and SQLColumns.
- Disabled (0): The driver does not return the hive_system table for catalog function calls.

Socket Timeout

Key Name	Default Value	Required
SocketTimeout	60	No

Description

The number of seconds that an operation can remain idle before it is closed.

> ✎ **Note:**
>
> This option is applicable only when asynchronous query execution is being used against Hive Server 2 instances.

Thrift Transport

Key Name	Default Value	Required
ThriftTransport	Binary (0) if you are connecting to Hive Server 1. SASL (1) if you are connecting to Hive Server 2.	No

Description

The transport protocol to use in the Thrift layer.

Select one of the following settings, or set the key to the number corresponding to the desired setting:

- Binary (0)
- SASL (1)
- HTTP (2)

> ✎ **Note:**
>
> For information about how to determine which Thrift transport protocols your Hive server supports, see Authentication Mechanisms on page 66.

Trusted Certificates

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

Description

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

If this option is not set, then the driver defaults to using the trusted CA certificates `.pem` file installed by the driver. To use the trusted CA certificates in the `.pem` file, set the `UseSystemTrustStore` property to `0` or clear the Use System Trust Store check box in the SSL Options dialog.

> ❗ **Important:**
>
> If you are connecting from a Windows machine and the Use System Trust Store option is enabled, the driver uses the certificates from the Windows trust store instead of your specified `.pem` file.
>
> For more information, see Use System Trust Store on page 102.

Two-Way SSL

Key Name	Default Value	Required
`TwoWaySSL`	Clear (`0`)	No

Description

This option specifies whether two-way SSL is enabled.

- Enabled (1): The client and the Hive server verify each other using SSL. See also the driver configuration options Client Certificate File on page 83, Client Private Key File on page 83, and Client Private Key Password on page 84.
- Disabled (0): The server does not verify the client. Depending on whether one-way SSL is enabled, the client might verify the server. For more information, see Enable SSL on page 87.

> ✎ **Note:**
>
> This option is applicable only when connecting to a Hive server that supports SSL. You must enable SSL before Two Way SSL can be configured. For more information, see Enable SSL on page 87.

Unicode SQL Character Types

Key Name	Default Value	Required
UseUnicodeSqlCharacterTypes	Clear (0)	No

Description

This option specifies the SQL types to be returned for string data types.

- Enabled (1): The driver returns SQL_WVARCHAR for STRING and VARCHAR columns, and returns SQL_WCHAR for CHAR columns.
- Disabled (0): The driver returns SQL_VARCHAR for STRING and VARCHAR columns, and returns SQL_CHAR for CHAR columns.

Use Async Exec

Key Name	Default Value	Required
EnableAsyncExec	Clear (0)	No

Description

This option specifies whether to execute queries synchronously or asynchronously.

- Enabled (`1`): The driver uses an asynchronous version of the API call against Hive for executing a query.
- Disabled (`0`): The driver executes queries synchronously.

Use Native Query

Key Name	Default Value	Required
UseNativeQuery	Clear (0)	No

Description

This option specifies whether the driver uses native HiveQL queries, or converts the queries emitted by an application into an equivalent form in HiveQL. If the application is Hive-aware and already emits HiveQL, then enable this option to avoid the extra overhead of query transformation.

- Enabled (`1`): The driver does not transform the queries emitted by an application, and executes HiveQL queries directly.
- Disabled (`0`): The driver transforms the queries emitted by an application and converts them into an equivalent form in HiveQL.

> **! Important:**
>
> When this option is enabled, the driver cannot execute parameterized queries.

Use Only SSPI

Key Name	Default Value	Required
UseOnlySSPI	Clear (0)	No

Description

This option specifies how the driver handles Kerberos authentication: either with the SSPI plugin or with MIT Kerberos.

- Enabled (`1`): The driver handles Kerberos authentication by using the SSPI plugin instead of MIT Kerberos by default.
- Disabled (`0`): The driver uses MIT Kerberos to handle Kerberos authentication, and only uses the SSPI plugin if the GSSAPI library is not available.

> **! Important:**
>
> This option is only available on Windows.

Use System Trust Store

Key Name	Default Value	Required
`UseSystemTrustStore`	Clear (`0`)	No

Description

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

- Enabled (`1`): The driver verifies the connection using a certificate in the system trust store.
- Disabled (`0`): The driver verifies the connection using a specified `.pem` file. For information about specifying a `.pem` file, see Trusted Certificates on page 99.

> **! Important:**
>
> If you are connecting from a Windows machine and you want to specify a `.pem` file containing trusted CA certificates, this option must be disabled. For more information, see Trusted Certificates on page 99.

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

User Name

Key Name	Default Value	Required
UID	For User Name (2) authentication only, the default value is `anonymous`	Yes, if the authentication mechanism is User Name And Password (3) or Windows Azure HDInsight Service (6). No, if the authentication mechanism is User Name (2).

Description

The user name that you use to access Hive Server 2.

ZK Force Active Passive HA

Key Name	Default Value	Required
ZKForceActivePassiveHA	0	No

Description

If the `ZkNamespace` property does not contain "ActivePassiveHA", this option forces the ZooKeeper discovery mode to Active Passive HS2 Interactive HA.

- Enabled (1): The driver enables the Active Passive HS2 Interactive HA mode.

- Disabled (0): The driver does not enable the Active Passive HS2 Interactive HA mode.

✎ **Note:**

This property is ignored if the `ZkNamespace` property contains "ActivePassiveHA".

ZooKeeper Namespace

Key Name	Default Value	Required
ZKNamespace	None	Yes, if Service Discovery Mode is enabled.

Description

The namespace on ZooKeeper under which Hive Server 2 znodes are added.

> ✎ **Note:**
>
> If the namespace contains "ActivePassiveHA", the driver will enable the Active Passive HS2 Interactive HA connection mode.

Configuration Options Having Only Key Names

The following configuration options do not appear in the Windows user interface for the Simba Hive ODBC Driver. They are accessible only when you use a connection string or configure a connection from a Linux/macOS/AIX/Solaris machine:

- ADUserNameCase on page 104
- DelegationUserIDCase on page 105
- Driver on page 106
- ForceSynchronousExec on page 106
- HTTPAuthCookies on page 107
- http.header. on page 107
- SSP_ on page 108

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

- UseLogPrefix on page 108

ADUserNameCase

Key Name	Default Value	Required
ADUserNameCase	Unchanged	No

Description

This option controls whether the driver changes the user name part of an AD Kerberos UPN to all upper-case or all lower-case. The following values are supported:

- `Upper`: Change the user name to all upper-case.
- `Lower`: Change the user name to all lower-case.
- `Unchanged`: Do not modify the user name.

> ✎ **Note:**
>
> This option is applicable only when using Active Directory Kerberos from a Windows client machine to authenticate.

DelegationUserIDCase

Key Name	Default Value	Required
DelegationUserIDCase	Unchanged	No

Description

This option specifies whether the driver changes the Delegation UID (or `DelegationUID`) value to all upper-case or all lower-case. The following values are supported:

- `Upper`: Change the delegated user name to all upper-case.
- `Lower`: Change the delegated user name to all lower-case.
- `Unchanged`: Do not modify the delegated user name.

For more information about delegating a user name, see Delegation UID on page 86.

Driver

Key Name	Default Value	Required
Driver	Simba Hive 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 Hive 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.

ForceSynchronousExec

Key Name	Default Value	Required
ForceSynchronousExec	0	No

Description

When this option is enabled (1), the driver is forced to execute queries synchronously when connected to an HDInsight cluster.

When this option is disabled (0), the driver is able to execute queries asynchronously when connected to an HDInsight cluster.

> ✎ **Note:**
>
> This option is applicable only to HDInsight clusters.

HTTPAuthCookies

Key Name	Default Value	Required
HTTPAuthCookies	hive.server2.auth, JSessionID	No

Description

A comma-separated list of authentication cookies that are supported by the driver.

If cookie-based authentication is enabled in your server, the driver authenticates the connection once based on the provided authentication credentials. It then uses the cookie generated by the server for each subsequent request in the same connection.

http.header.

Key Name	Default Value	Required
http.header.	None	No

Description

Set a custom HTTP header by using the following syntax, where *[HeaderKey]* is the name of the header to set and *[HeaderValue]* is the value to assign to the header:

```
http.header.[HeaderKey]=[HeaderValue]
```

For example:

```
http.header.AUTHENTICATED_USER=john
```

After the driver applies the header, the http.header. prefix is removed from the DSN entry, leaving an entry of *[HeaderKey]=[HeaderValue]*

The example above would create the following custom HTTP header:

```
AUTHENTICATED_USER: john
```

> ✎ **Note:**
>
> - The `http.header.` prefix is case-sensitive.
> - This option is applicable only when you are using HTTP as the Thrift transport protocol. For more information, see Thrift Transport on page 98.

SSP_

Key Name	Default Value	Required
`SSP_`	None	No

Description

Set a server-side property by using the following syntax, where *[SSPKey]* is the name of the server-side property and *[SSPValue]* is the value for that property:

```
SSP_[SSPKey]=[SSPValue]
```

For example:

```
SSP_mapred.queue.names=myQueue
```

After the driver applies the server-side property, the `SSP_` prefix is removed from the DSN entry, leaving an entry of *[SSPKey]=[SSPValue]*.

> ✎ **Note:**
>
> - The `SSP_` prefix must be upper case.
> - When setting a server-side property in a connection string, it is recommended that you enclose the value in braces (`{ }`) to make sure that special characters can be properly escaped.

UseLogPrefix

Key Name	Default Value	Required
`UseLogPrefix`	0	No

Description

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

Set the property to one of the following values:

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

  For example, if you are connecting as a user named "jdoe" and using the driver in an application with process ID 7836, the generated log files would be named `jdoe_7836_simbahiveodbcdriver.log` and `jdoe_7836_simbahiveodbcdriver_connection_[Number].log`, where *[Number]* is a number that identifies each connection-specific log file.

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

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

- For a 32-bit driver installed on a 64-bit machine: **HKEY_LOCAL_ MACHINE\SOFTWARE\Wow6432Node\Simba\Simba Hive ODBC Driver\Driver**
- Otherwise: **HKEY_LOCAL_MACHINE\SOFTWARE\Simba\Simba Hive ODBC Driver\Driver**

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

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

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.

IBM and AIX are trademarks or registered trademarks of IBM Corporation or its subsidiaries in Canada, United States, and/or other countries.

Kerberos is a trademark of the Massachusetts Institute of Technology (MIT).

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 Azure, 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.

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

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.

Apache Hive, Apache, and Hive are trademarks or registered trademarks of The Apache Software Foundation or its subsidiaries in Canada, United States and/or other countries.

All other trademarks are trademarks of their respective owners.