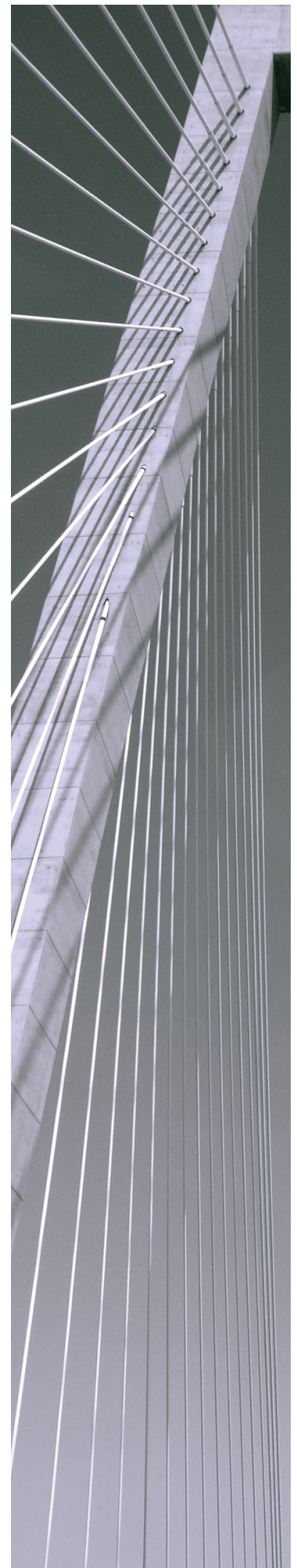# Simba Impala JDBC Driver with SQL Connector

Installation and Configuration Guide

Simba Technologies Inc.

Version 1.0.46

March 8, 2017

Contact Us

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

Tel: +1 (604) 633-0008

Fax: +1 (604) 633-0004

www.simba.com

About This Guide

Purpose

The *Simba Impala JDBC Driver with SQL Connector Installation and Configuration Guide* explains how to install and configure the Simba Impala JDBC Driver with SQL Connector on all supported platforms. The guide also provides details related to features of the driver.

Audience

The guide is intended for end users of the Simba Impala JDBC Driver.

Knowledge Prerequisites

To use the Simba Impala JDBC Driver, the following knowledge is helpful:

- Familiarity with the platform on which you are using the Simba Impala JDBC Driver
- Ability to use the data store to which the Simba Impala JDBC Driver is connecting
- An understanding of the role of JDBC technologies in connecting to a data store
- Experience creating and configuring JDBC 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 Impala JDBC Driver

The Simba Impala JDBC Driver is used for direct SQL and Impala SQL access to Apache Hadoop / Impala distributions, enabling Business Intelligence (BI), analytics, and reporting on Hadoop / Impala-based data. The driver efficiently transforms an application's SQL query into the equivalent form in Impala SQL, which is a subset of SQL-92. If an application is Impala-aware, then the driver is configurable to pass the query through to the database for processing. The driver interrogates Impala to obtain schema information to present to a SQL-based application. Queries, including joins, are translated from SQL to Impala SQL. For more information about the differences between Impala SQL and SQL, see Features on page 29.

The Simba Impala JDBC Driver complies with the JDBC 3.0, 4.0, and 4.1 data standards. JDBC 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 JDBC driver, which connects an application to the database. For more information about JDBC, see the *Data Access Standards Glossary*:
http://www.simba.com/resources/data-access-standards-library.

This guide is suitable for users who want to access data residing within Impala from their desktop environment. Application developers might also find the information helpful. Refer to your application for details on connecting via JDBC.

System Requirements

Each machine where you use the Simba Impala JDBC Driver must have Java Runtime Environment (JRE) installed. The version of JRE that must be installed depends on the version of the JDBC API you are using with the driver. The following table lists the required version of JRE for each version of the JDBC API.

JDBC API Version	JRE Version
3.0	4.0 or 5.0
4.0	6.0 or later
4.1	7.0 or later

The driver supports Cloudera Impala versions 1.0.1 through 2.8.

Simba Impala JDBC Driver Files

The Simba Impala JDBC Driver is delivered in the following two ZIP archives, where *[Version]* is the version number of the driver:

- `ImpalaJDBC3_[Version].zip`
- `ImpalaJDBC4_[Version].zip`
- `ImpalaJDBC41_[Version].zip`

Each archive contains the driver supporting the JDBC API version indicated in the archive name, as well as release notes and third party license information.

Installing and Using the Simba Impala JDBC Driver

To install the Simba Impala JDBC Driver on your machine, extract the files from the appropriate ZIP archive to the directory of your choice.

> ❗ **Important:**
>
> If you received a license file through email, then you must copy the file into the same directory as the `ImpalaJDBC3.jar`, `ImpalaJDBC4.jar`, or `ImpalaJDBC41.jar` file before you can use the Simba Impala JDBC Driver.

To access a Impala data store using the Simba Impala JDBC Driver, you need to configure the following:

- The list of driver library files (see Referencing the JDBC Driver Libraries on page 10)
- The Driver or DataSource class (see Registering the Driver Class on page 11)
- The connection URL for the driver (see Building the Connection URL on page 12)

> ❗ **Important:**
>
> The Simba Impala JDBC Driver provides read-only access to Impala data.

Referencing the JDBC Driver Libraries

Before you use the Simba Impala JDBC Driver, the JDBC application or Java code that you are using to connect to the data store must be able to access the driver JAR files. In the application or code, specify all the JAR files that you extracted from the appropriate ZIP archive.

Using the Driver in a JDBC Application

Most JDBC applications provide a set of configuration options for adding a list of driver library files. Use the provided options to include all the JAR files from the ZIP archive as part of the driver configuration in the application. For more information, see the documentation for your JDBC application.

Using the Driver in Java Code

You must include all the driver library files in the class path. This is the path that the Java Runtime Environment searches for classes and other resource files. For more information, see "Setting the Class Path" in the Java SE Documentation:

- For Windows:
  http://docs.oracle.com/javase/7/docs/technotes/tools/windows/classpath.html
- For Linux and Solaris:
  http://docs.oracle.com/javase/7/docs/technotes/tools/solaris/classpath.html

Registering the Driver Class

Before connecting to the data store, you must register the appropriate class for your application.

The following is a list of the classes used to connect the Simba Impala JDBC Driver to Impala data stores. The `Driver` classes extend `java.sql.Driver`, and the `DataSource` classes extend `javax.sql.DataSource` and `javax.sql.ConnectionPoolDataSource`.

To support JDBC 3.0, classes with the following fully-qualified class names (FQCNs) are available:

- `com.simba.impala.jdbc3.Driver`
- `com.simba.impala.jdbc3.DataSource`

To support JDBC 4.0, classes with the following FQCNs are available:

- `com.simba.impala.jdbc4.Driver`
- `com.simba.impala.jdbc4.DataSource`

To support JDBC 4.1, classes with the following FQCNs are available:

- `com.simba.impala.jdbc41.Driver`
- `com.simba.impala.jdbc41.DataSource`

The following sample code shows how to use the `DriverManager` to establish a connection for JDBC 4:

> ✎ **Note:**
>
> In these examples, the line `Class.forName(DRIVER_CLASS);` is only required for JDBC 4.0 and earlier.

```
private static Connection connectViaDM() throws Exception
{
    Connection connection = null;
    Class.forName(DRIVER_CLASS);
    connection = DriverManager.getConnection(CONNECTION_URL);
    return connection;
```

```
}
```

The following sample code shows how to use the `DataSource` class to establish a connection:

```
private static Connection connectViaDS() throws Exception
{
    Connection connection = null;
    Class.forName(DRIVER_CLASS);
    DataSource ds = new com.simba.impala.jdbc41.DataSource();
    ds.setURL(CONNECTION_URL);
    connection = ds.getConnection();
    return connection;
}
```

Building the Connection URL

Use the connection URL to supply connection information to the data store that you are accessing. The following is the format of the connection URL for the Simba Impala JDBC Driver, where *[Host]* is the DNS or IP address of the Impala server and *[Port]* is the number of the TCP port that the server uses to listen for client requests:

```
jdbc:impala://[Host]:[Port]
```

> ✎ **Note:**
>
> By default, Impala uses port 21050.

By default, the driver uses the schema named **default**.

You can specify optional settings such as the schema to use or any of the connection properties supported by the driver. For a list of the properties available in the driver, see Driver Configuration Options on page 32.

> ✎ **Note:**
>
> If you specify a property that is not supported by the driver, then the driver attempts to apply the property as a Impala server-side property for the client session. For more information, see Configuring Server-Side Properties on page 26.

The following is the format of a connection URL that specifies some optional settings:

```
jdbc:impala://[Host]:[Port]/[Schema];[Property1]=[Value];
[Property2]=[Value];...
```

For example, to connect to port 18000 on an Impala server installed on the local machine, use a schema named default2, and authenticate the connection using a user name and password, you would use the following connection URL:

```
jdbc:impala://node1.example.com:18000/default2;AuthMech=3;
UID=simba;PWD=simba
```

❗ **Important:**

- Properties are case-sensitive.
- Do not duplicate properties in the connection URL.

Configuring Authentication

The Simba Impala JDBC Driver supports the following authentication mechanisms:

- No Authentication
- Kerberos
- User Name
- User Name And Password

You configure the authentication mechanism that the driver uses to connect to Impala by specifying the relevant properties in the connection URL.

For information about configuring the authentication mechanism that Impala uses, see the Impala documentation:
http://www.cloudera.com/content/cloudera/en/documentation.html.

For information about the properties you can use in the connection URL, see Driver Configuration Options on page 32.

> ✎ **Note:**
>
> In addition to authentication, you can configure the driver to connect over SSL. For more information, see Configuring SSL on page 24.

Using No Authentication

You provide this information to the driver in the connection URL. For more information about the syntax of the connection URL, see Building the Connection URL on page 12.

To configure a connection without authentication:

➤ Set the `AuthMech` property to `0`.

For example:

```
jdbc:impala://localhost:21050;AuthMech=0;
```

Using Kerberos

Kerberos must be installed and configured before you can use this authentication mechanism. For information about configuring and operating Kerberos on Windows, see Configuring Kerberos Authentication for Windows on page 16. For other operating

systems, see the MIT Kerberos documentation: http://web.mit.edu/kerberos/krb5-latest/doc/.

You provide this information to the driver in the connection URL. For more information about the syntax of the connection URL, see Building the Connection URL on page 12.

> **✎ Note:**
>
> The driver also supports Kerberos constrained delegation. For more details on this, see Using Kerberos Constrained Delegation on page 23.

To configure default Kerberos authentication:

1. Set the `AuthMech` property to `1`.
2. To use the default realm defined in your Kerberos setup, do not set the `KrbRealm` property.

   If your Kerberos setup does not define a default realm or if the realm of your Impala server is not the default, then set the `KrbRealm` property to the realm of the Impala server.
3. Set the `KrbHostFQDN` property to the fully qualified domain name of the Impala server host.
4. If you are using Kerberos Constrained Delegation, set the `userGSSCredential` property to your Kerberos GSS Credential.
5. Optionally, specify how the driver obtains the Kerberos Subject by setting the `KrbAuthType` property as follows:
   - To configure the driver to automatically detect which method to use for obtaining the Subject, set the `KrbAuthType` property to `0`. Alternatively, do not set the `KrbAuthType` property.
   - Or, to create a LoginContext from a JAAS configuration and then use the Subject associated with it, set the `KrbAuthType` property to `1`.
   - Or, to create a LoginContext from a Kerberos ticket cache and then use the Subject associated with it, set the `KrbAuthType` property to `2`.

   For more detailed information about how the driver obtains Kerberos Subjects based on these settings, see KrbAuthType on page 35.

For example:

```
jdbc:impala://node1.example.com:21050;AuthMech=1;
KrbRealm=EXAMPLE.COM;KrbHostFQDN=node1.example.com;
KrbServiceName=impala
```

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.

You provide this information to the driver in the connection URL. For more information about the syntax of the connection URL, see Building the Connection URL on page 12.

To configure User Name authentication:

1. Set the `AuthMech` property to `2`.
2. Set the `UID` property to an appropriate user name for accessing the Impala server.

For example:

```
jdbc:impala://node1.example.com:21050;AuthMech=2;UID=impala
```

Using User Name And Password

This authentication mechanism requires a user name and a password.

You provide this information to the driver in the connection URL. For more information about the syntax of the connection URL, see Building the Connection URL on page 12.

To configure User Name And Password authentication:

1. Set the `AuthMech` property to `3`.
2. Set the `UID` property to an appropriate user name for accessing the Impala server.
3. Set the `PWD` property to the password corresponding to the user name you provided.

For example:

```
jdbc:impala://node1.example.com:21050;AuthMech=3;
UID=impala;PWD=simba;UseSasl=0;
```

Configuring Kerberos Authentication for Windows

You can configure your Kerberos setup so that you use the MIT Kerberos Ticket Manager to get the Ticket Granting Ticket (TGT), or configure the setup so that you can use the driver to get the ticket directly from the Key Distribution Center (KDC). Also, if a client application obtains a Subject with a TGT, it is possible to use that Subject to authenticate the connection.

Downloading and Installing MIT Kerberos for Windows

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.
3. Follow the instructions in the installer to complete the installation process.
4. When the installation completes, click **Finish**.

Using the MIT Kerberos Ticket Manager to Get Tickets

Setting the KRB5CCNAME Environment Variable

You must set the KRB5CCNAME environment variable to your credential cache file.

To set the KRB5CCNAME environment variable:

1. Click **Start** 🌐, then right-click **Computer**, and then click **Properties**.
2. Click **Advanced System Settings**.
3. In the System Properties dialog box, on the **Advanced** tab, click **Environment Variables**.
4. In the Environment Variables dialog box, under the System Variables list, click **New**.
5. In the **New System Variable** dialog box, in the Variable Name field, type **KRB5CCNAME**.
6. In the **Variable Value** field, type the path for your credential cache file. For example, type `C:\KerberosTickets.txt`.
7. Click **OK** to save the new variable.
8. Make sure that the variable appears in the System Variables list.
9. Click **OK** to close the Environment Variables dialog box, and then click **OK** to

close the System Properties dialog box.

10. Restart your machine.

Getting a Kerberos Ticket

To get a Kerberos ticket:

1. Click **Start** , then click **All Programs**, and then click the **Kerberos for Windows (64-bit)** or **Kerberos for Windows (32-bit)** program group.
2. Click **MIT Kerberos Ticket Manager**.
3. In the MIT Kerberos Ticket Manager, click **Get Ticket**.
4. 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 the MIT Kerberos Ticket Manager.

Authenticating to the Impala Server

You provide this information to the driver in the connection URL. For more information about the syntax of the connection URL, see Building the Connection URL on page 12.

To authenticate to the Impala server:

➢ Use a connection URL that has the following properties defined:

- AuthMech
- KrbHostFQDN
- KrbRealm
- KrbServiceName

For detailed information about these properties, see Driver Configuration Options on page 32

Using the Driver to Get Tickets

Deleting the KRB5CCNAME Environment Variable

To enable the driver to get Ticket Granting Tickets (TGTs) directly, make sure that the KRB5CCNAME environment variable has not been set.

To delete the KRB5CCNAME environment variable:

1. Click the **Start** button , then right-click **Computer**, and then click **Properties**.
2. Click **Advanced System Settings**.

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

4. In the Environment Variables dialog box, check if the KRB5CCNAME variable appears in the System variables list. If the variable appears in the list, then select the variable and click **Delete**.

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

Setting Up the Kerberos Configuration File

To set up the Kerberos configuration file:

1. Create a standard `krb5.ini` file and place it in the `C:\Windows` directory.

2. Make sure that the KDC and Admin server specified in the `krb5.ini` file can be resolved from your terminal. If necessary, modify `C:\Windows\System32\drivers\etc\hosts`.

Setting Up the JAAS Login Configuration File

To set up the JAAS login configuration file:

1. Create a JAAS login configuration file that specifies a keytab file and `doNotPrompt=true`.

   For example:

   ```
   Client {
   com.sun.security.auth.module.Krb5LoginModule required
   useKeyTab=true
   keyTab="PathToTheKeyTab"
   principal="simba@SIMBA"
   doNotPrompt=true;
   };
   ```

2. Set the `java.security.auth.login.config` environment variable to the location of the JAAS file.

   For example: `C:\KerberosLoginConfig.ini`.

Authenticating to the Impala Server

You provide this information to the driver in the connection URL. For more information about the syntax of the connection URL, see Building the Connection URL on page 12.

To authenticate to the Impala server:

➢ Use a connection URL that has the following properties defined:

- `AuthMech`
- `KrbHostFQDN`
- `KrbRealm`
- `KrbServiceName`

For detailed information about these properties, see Driver Configuration Options on page 32.

Using an Existing Subject to Authenticate the Connection

If the client application obtains a Subject with a TGT, then that Subject can be used to authenticate the connection to the server.

To use an existing Subject to authenticate the connection:

1. Create a PrivilegedAction for establishing the connection to the database.

   For example:

```
// Contains logic to be executed as a privileged action
public class AuthenticateDriverAction
implements PrivilegedAction<Void>
{
// The connection, which is established as a
PrivilegedAction
Connection con;

// Define a string as the connection URL
static String ConnectionURL =
"jdbc:impala://192.168.1.1:21050";

/**
* Logic executed in this method will have access to the
* Subject that is used to "doAs". The driver will get
* the Subject and use it for establishing a connection
* with the server.
*/
@Override
public Void run()
{
try
```

```
{
// Establish a connection using the connection URL
con = DriverManager.getConnection(ConnectionURL);
}
catch (SQLException e)
{
// Handle errors that are encountered during
// interaction with the data store
e.printStackTrace();
}
catch (Exception e)
{
// Handle other errors
e.printStackTrace();
}
return null;
}
}
```

2. Run the PrivilegedAction using the existing Subject, and then use the connection.

For example:

```
// Create the action
AuthenticateDriverAction authenticateAction = new
AuthenticateDriverAction();
// Establish the connection using the Subject for
// authentication.
Subject.doAs(loginConfig.getSubject(),
authenticateAction);
// Use the established connection.
authenticateAction.con;
```

Kerberos Encryption Strength and the JCE Policy Files Extension

If the encryption being used in your Kerberos environment is too strong, you might encounter the error message "Unable to connect to server: GSS initiate failed" when trying to use the driver to connect to a Kerberos-enabled cluster. Typically, Java vendors only allow encryption strength up to 128 bits by default. If you are using

greater encryption strength in your environment (for example, 256-bit encryption), then you might encounter this error.

Diagnosing the Issue

If you encounter the error message "Unable to connect to server: GSS initiate failed", confirm that it is occurring due to encryption strength by enabling Kerberos layer logging in the JVM and then checking if the log output contains the error message "KrbException: Illegal key size".

To enable Kerberos layer logging in a Sun JVM:

➢ Choose one:

- In the Java command you use to start the application, pass in the following argument:

```
-Dsun.security.krb5.debug=true
```

- Or, add the following code to the source code of your application:

```
System.setProperty("sun.security.krb5.debug","true")
```

To enable Kerberos layer logging in an IBM JVM:

➢ Choose one:

- In the Java command you use to start the application, pass in the following arguments:

```
-Dcom.ibm.security.krb5.Krb5Debug=all
-Dcom.ibm.security.jgss.debug=all
```

- Or, add the following code to the source code of your application:

```
System.setProperty
("com.ibm.security.krb5.Krb5Debug","all");
System.setProperty
("com.ibm.security.jgss.debug","all");
```

Resolving the Issue

After you confirm that the error is occurring due to encryption strength, you can resolve the issue by downloading and installing the *Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files* extension from your Java vendor. Refer to the instructions from the vendor to install the files to the correct location.

> **! Important:**
>
> Consult your company's policy to make sure that you are allowed to enable encryption strengths in your environment that are greater than what the JVM allows by default.

If the issue is not resolved after you install the JCE policy files extension, then restart your machine and try your connection again. If the issue persists even after you restart your machine, then verify which directories the JVM is searching to find the JCE policy files extension. To print out the search paths that your JVM currently uses to find the JCE policy files extension, modify your Java source code to print the return value of the following call:

```
System.getProperty("java.ext.dirs")
```

Using Kerberos Constrained Delegation

The driver can also be configured to use Kerberos Constrained Delegation. This feature allows a service to obtain service tickets to a restricted list of other services running on specific servers on the network after it has been presented with a service ticket. For more details on the process see: https://technet.microsoft.com/en-ca/library/cc995228.aspx.

The `userGSSCredential` connection property can be used in the connection URL to pass in a GSSCredential object. The following sample code shows how to use the property to pass the GSSCredential into the driver using JDBC 4.1.

```
GSSCredential userCredential = [GSSCredential]
Driver driver = (Driver) Class.forName
("com.simba.impala.jdbc41.Driver").newInstance();
Properties properties = new Properties();
properties.put("userGSSCredential", userCredential);
Connection conn = driver.connect(

   "jdbc:impala://node1.example.com:21050;AuthMech=1;KrbReal
   m=EXAMPLE.COM;
   KrbHostFQDN=node1.example.com;KrbServiceName=impala"
   ,properties);
```

Configuring SSL

> ✏ **Note:**
>
> In this documentation, "SSL" indicates both TLS (Transport Layer Security) and SSL (Secure Sockets Layer). The driver supports industry-standard versions of TLS/SSL.

If you are connecting to an Impala server that has Secure Sockets Layer (SSL) enabled, you can configure the driver to connect to an SSL-enabled socket. When connecting to a server over SSL, the driver uses one-way authentication to verify the identity of the server.

One-way authentication requires a signed, trusted SSL certificate for verifying the identity of the server. You can configure the driver to access a specific TrustStore or KeyStore that contains the appropriate certificate. If you do not specify TrustStore or KeyStore, then the driver uses the default Java TrustStore named `jssecacerts`. If `jssecacerts` is not available, then the driver uses `cacerts` instead.

You provide this information to the driver in the connection URL. For more information about the syntax of the connection URL, see Building the Connection URL on page 12.

To configure SSL:

1. If you are not using one of the default Java TrustStores, then do one of the following:
    - Create a TrustStore and configure the driver to use it:
        a. Create a TrustStore containing your signed, trusted server certificate.
        b. Set the `SSLTrustStore` property to the full path of the TrustStore.
        c. Set the `SSLTrustStorePwd` property to the password for accessing the TrustStore.
    - Or, create a KeyStore and configure the driver to use it:
        a. Create a KeyStore containing your signed, trusted server certificate.
        b. Set the `SSLKeyStore` property to the full path of the KeyStore.
        c. Set the `SSLKeyStorePwd` property to the password for accessing the KeyStore.
2. Set the `SSL` property to `1`.
3. Optionally, to allow the SSL certificate used by the server to be self-signed, set the `AllowSelfSignedCerts` property to `1`.

4. Optionally, to allow the common name of a CA-issued certificate to not match the host name of the Impala server, set the `CAIssuedCertNamesMismatch` property to `1`.

> **Note:**
>
> For self-signed certificates, the driver always allows the common name of the certificate to not match the host name.

For example:

```
jdbc:impala://localhost:21050;AuthMech=3;SSL=1;
SSLKeyStore=C:\\Users\\bsmith\\Desktop\\keystore.jks;
SSLKeyStorePwd=simbaSSL123;UID=impala;PWD=simba123
```

> **Note:**
>
> For more information about the connection properties used in SSL connections, see Driver Configuration Options on page 32

Configuring Server-Side Properties

When connecting to a server that is running Impala 2.0 or later, you can use the driver to apply configuration properties to the server by setting the properties in the connection URL.

> ❗ **Important:**
>
> This feature is not supported for earlier versions of Impala, where the SET statement can only be executed from within the Impala shell.

For example, to set the `MEM_LIMIT` query option to 1 GB and the `REQUEST_POOL` query option to `myPool`, you would use a connection URL such as the following:

```
jdbc:impala://localhost:18000/default2;AuthMech=3;
UID=simba;PWD=simba;MEM_LIMIT=1000000000;REQUEST_POOL=myPool
```

Configuring Logging

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.

In the connection URL, set the `LogLevel` key to enable logging at the desired level of detail. The following table lists the logging levels provided by the Simba Impala JDBC Driver, in order from least verbose to most verbose.

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

To enable logging:

1. Set the `LogLevel` property to the desired level of information to include in log files.
2. Set the `LogPath` property to the full path to the folder where you want to save log files. To make sure that the connection URL is compatible with all JDBC applications, escape the backslashes (\) in your file paths by typing another backslash.

For example, the following connection URL enables logging level 3 and saves the log files in the `C:\temp` folder:

```
jdbc:impala://localhost:11000;LogLevel=3;LogPath=C:\\temp
p
```

3. To make sure that the new settings take effect, restart your JDBC application and reconnect to the server.

   The Simba Impala JDBC Driver produces the following log files in the location specified in the `LogPath` property:

   - A `ImpalaJDBC_driver.log` file that logs driver activity that is not specific to a connection.
   - A `Impala_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 the `LogPath` value is invalid, then the driver sends the logged information to the standard output stream (`System.out`).

To disable logging:

1. Remove the `LogLevel` and `LogPath` properties from the connection URL.
2. To make sure that the new settings take effect, restart your JDBC application and reconnect to the server.

Features

More information is provided on the following features of the Simba Impala JDBC Driver:

- SQL Translation on page 29
- Data Types on page 29
- Catalog and Schema Support on page 30
- Security and Authentication on page 31

SQL Translation

The Simba Impala JDBC Driver is able to parse queries locally prior to sending them to the Impala server. This feature allows the driver to calculate query metadata without executing the query, support query parameters, and support extra SQL features such as JDBC escape sequences and additional scalar functions that are not available in the Impala-shell tool.

> ✎ **Note:**
>
> The driver does not support translation for queries that reference a field contained in a nested column (an ARRAY, MAP, or STRUCT column). To retrieve data from a nested column, make sure that the query is written in valid Impala SQL syntax.

Data Types

The Simba Impala JDBC Driver supports many common data formats, converting between Impala, SQL, and Java data types.

The following table lists the supported data type mappings.

Impala Type	SQL Type	Java Type
ARRAY	VARCHAR	String
BIGINT	BIGINT	java.math.BigInteger
BINARY	VARBINARY	byte[]
BOOLEAN	BOOLEAN	Boolean

Impala Type	SQL Type	Java Type
CHAR (Available only in CDH 5.2 or later)	CHAR	String
DATE	DATE	java.sql.Date
DECIMAL (Available only in CDH 5.1 or later)	DECIMAL	java.math.BigDecimal
DOUBLE (REAL is an alias for DOUBLE)	DOUBLE	Double
FLOAT	REAL	Float
INT	INTEGER	Long
MAP	VARCHAR	String
SMALLINT	SMALLINT	Integer
STRUCT	VARCHAR	String
TIMESTAMP	TIMESTAMP	java.sql.Timestamp
TINYINT	TINYINT	Short
VARCHAR (Available only in CDH 5.2 or later)	VARCHAR	String

Catalog and Schema Support

The Simba Impala JDBC Driver supports both catalogs and schemas to make it easy for the driver to work with various JDBC applications. Since Impala only organizes tables into schemas/databases, the driver provides a synthetic catalog named IMPALA

under which all of the schemas/databases are organized. The driver also maps the JDBC schema to the Impala schema/database.

> ✎ **Note:**
>
> Setting the `CatalogSchemaSwitch` connection property to `1` will cause Impala catalogs to be treated as schemas in the driver as a restriction for filtering.

Security and Authentication

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

> ✎ **Note:**
>
> In this documentation, "SSL" indicates both TLS (Transport Layer Security) and SSL (Secure Sockets Layer). The driver supports industry-standard versions of TLS/SSL.

The driver provides mechanisms that allow you to authenticate your connection using the Kerberos protocol, your Impala user name only, or your Impala user name and password. You must use the authentication mechanism that matches the security requirements of the Impala server. For detailed driver configuration instructions, see Configuring Authentication on page 14.

Additionally, the driver supports SSL connections with one-way authentication. If the server has an SSL-enabled socket, then you can configure the driver to connect to it.

It is recommended that you enable SSL whenever you connect to a server that is configured to support it. SSL encryption protects data and credentials when they are transferred over the network, and provides stronger security than authentication alone. For detailed configuration instructions, see Configuring SSL on page 24.

The SSL version that the driver supports depends on the JVM version that you are using. For information about the SSL versions that are supported by each version of Java, see "Diagnosing TLS, SSL, and HTTPS" on the Java Platform Group Product Management Blog: https://blogs.oracle.com/java-platform-group/entry/diagnosing_tls_ssl_and_https.

> ✎ **Note:**
>
> The SSL version used for the connection is the highest version that is supported by both the driver and the server, which is determined at connection time.

Driver Configuration Options

Driver Configuration Options lists and describes the properties that you can use to configure the behavior of the Simba Impala JDBC Driver.

You can set configuration properties using the connection URL. For more information, see Building the Connection URL on page 12.

> ✎ **Note:**
>
> Property names and values are case-sensitive.

AllowSelfSignedCert

Default Value	Data Type	Required
0	Integer	No

Description

This property specifies whether the driver allows the server to use self-signed SSL certificates.

- `1`: The driver allows self-signed certificates.
- `0`: The driver does not allow self-signed certificates.

> ✎ **Note:**
>
> This property is applicable only when SSL connections are enabled.

AsyncExecPollInterval

Default Value	Data Type	Required
10	Integer	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 Impala is asynchronous. It does not mean that JDBC asynchronous operations are supported.

AuthMech

Default Value	Data Type	Required
0	Integer	No

Description

The authentication mechanism to use. Set the property to one of the following values:

- `0` for No Authentication.
- `1` for Kerberos.
- `2` for User Name.
- `3` for User Name And Password.

CAIssuedCertsMismatch

Default Value	Data Type	Required
0	Integer	No

Description

This property specifies whether the driver requires the name of the CA-issued SSL certificate to match the host name of the Impala server.

- `0`: The driver requires the names to match.
- `1`: The driver allows the names to mismatch.

> ✎ **Note:**
>
> This property is applicable only when SSL connections are enabled.

CatalogSchemaSwitch

Default Value	Data Type	Required
0	Integer	No

Description

This property specifies whether the driver treats Impala catalogs as schemas or as catalogs.

- `1`: The driver treats Impala catalogs as schemas as a restriction for filtering.
- `0`: Impala catalogs are treated as catalogs, and Impala schemas are treated as schemas.

DefaultStringColumnLength

Default Value	Data Type	Required
255	Integer	No

Description

The maximum number of characters that can be contained in STRING columns. The range of `DefaultStringColumnLength` is `0` to `32767`.

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

DelegationUID

Default Value	Data Type	Required
None	String	No

Description

Use this option to delegate all operations against Impala to a user that is different than the authenticated user for the connection.

KrbAuthType

Default Value	Data Type	Required
0	Integer	No

Description

This property specifies how the driver obtains the Subject for Kerberos authentication.

- `0`: The driver automatically detects which method to use for obtaining the Subject:
  1. First, the driver tries to obtain the Subject from the current thread's inherited AccessControlContext. If the AccessControlContext contains multiple Subjects, the driver uses the most recent Subject.
  2. If the first method does not work, then the driver checks the `java.security.auth.login.config` system property for a JAAS configuration. If a JAAS configuration is specified, the driver uses that information to create a LoginContext and then uses the Subject associated with it.
  3. If the second method does not work, then the driver checks the KRB5_CONFIG and KRB5CCNAME system environment variables for a Kerberos ticket cache. The driver uses the information from the cache to create a LoginContext and then uses the Subject associated with it.
- `1`: The driver checks the `java.security.auth.login.config` system property for a JAAS configuration. If a JAAS configuration is specified, the driver uses that information to create a LoginContext and then uses the Subject associated with it.
- `2`: The driver checks the KRB5_CONFIG and KRB5CCNAME system environment variables for a Kerberos ticket cache. The driver uses the information from the cache to create a LoginContext and then uses the Subject associated with it.

KrbHostFQDN

Default Value	Data Type	Required
None	String	Yes, if `AuthMech=1`.

Description

The fully qualified domain name of the Impala host.

KrbRealm

Default Value	Data Type	Required
Depends on your Kerberos configuration.	String	No

Description

The realm of the Impala host.

If your Kerberos configuration already defines the realm of the Impala host as the default realm, then you do not need to configure this property.

KrbServiceName

Default Value	Data Type	Required
None	String	Yes, if `AuthMech=1.`

Description

The Kerberos service principal name of the Impala server.

LogLevel

Default Value	Data Type	Required
0	Integer	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.

Set the property to one of the following numbers:

- `0`: Disable all logging.
- `1`: Enable logging on the FATAL level, which logs very severe error events that will lead the driver to abort.
- `2`: Enable logging on the ERROR level, which logs error events that might still allow the driver to continue running.
- `3`: Enable logging on the WARNING level, which logs events that might result in an error if action is not taken.
- `4`: Enable logging on the INFO level, which logs general information that describes the progress of the driver.
- `5`: Enable logging on the DEBUG level, which logs detailed information that is useful for debugging the driver.
- `6`: Enable logging on the TRACE level, which logs all driver activity.

When logging is enabled, the driver produces the following log files in the location specified in the `LogPath` property:

- A `ImpalaJDBC_driver.log` file that logs driver activity that is not specific to a connection.
- A `Impala_connection_[Number].log` file for each connection made to the database, where *[Number]* is a number that distinguishes each log file from the others. This file logs driver activity that is specific to the connection.

If the `LogPath` value is invalid, then the driver sends the logged information to the standard output stream (`System.out`).

LogPath

Default Value	Data Type	Required
The current working directory	String	No

Description

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

LowerCaseResultSetColumnName

Default Value	Data Type	Required
1	Integer	No

Description

This property specifies the letter case that the driver uses when returning the column name aliases in the `ResultSetMetadata`.

- `1`: The column name aliases in the `ResultSetMetadata` are returned in lower-case characters, matching the server-side behavior.
- `0`: The column name aliases are returned in the same letter case as specified in the query.

OptimizedInsert

Default Value	Data Type	Required
1	Integer	No

Description

This property specifies whether the driver tries to optimize INSERT statements by bypassing translation.

Each time the driver translates an INSERT statement, it executes the DESCRIBE command to identify the data types of the columns that it is inserting data into. These additional commands consume resources and might reduce driver performance.

- `1`: The driver tries to optimize INSERT statements by bypassing translation and using other methods to identify column types.
- `0`: The driver does not attempt the optimization, and translates INSERT statements normally.

> ✎ **Note:**
>
> If the optimization fails, the driver falls back to translating INSERT statements normally. This additional overhead might further reduce driver performance.

PreparedMetaLimitZero

Default Value	Data Type	Required
1	Integer	No

Description

This property specifies whether the `PreparedStatement.getMetadata()` call will request metadata from the server with `LIMIT 0`, increasing performance.

- `1`: The `PreparedStatement.getMetadata()` call uses `LIMIT 0`.
- `0`: The `PreparedStatement.getMetadata()` call does not use `LIMIT 0`.

PWD

Default Value	Data Type	Required
None	String	Yes, if `AuthMech=3`.

Description

The password corresponding to the user name that you provided using the property .

RowsFetchedPerBlock

Default Value	Data Type	Required
10000	Integer	No

Description

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

Any positive 32-bit integer is a valid value, but testing has shown that performance gains are marginal beyond the default value of 10000 rows.

SocketTimeout

Default Value	Data Type	Required
30	Integer	No

Description

The number of seconds after which Impala closes the connection with the client application if the connection is idle.

When this property is set to 0, idle connections are not closed.

SSL

Default Value	Data Type	Required
0	Integer	No

Description

This property specifies whether the driver communicates with the Impala server through an SSL-enabled socket.

- 1: The driver connects to SSL-enabled sockets.
- 0: The driver does not connect to SSL-enabled sockets.

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

SSLKeyStore

Default Value	Data Type	Required
None	String	No

Description

The full path of the Java KeyStore containing the server certificate for one-way SSL authentication.

See also the property SSLKeyStorePwd on page 41.

> ✎ **Note:**
>
> The Simba Impala JDBC Driver accepts TrustStores and KeyStores for one-way SSL authentication. See also the property SSLTrustStore on page 41.

SSLKeyStorePwd

Default Value	Data Type	Required
None	Integer	Yes, if you are using a KeyStore for connecting over SSL.

Description

The password for accessing the Java KeyStore that you specified using the property SSLKeyStore on page 40.

SSLTrustStore

Default Value	Data Type	Required
`jssecacerts`, if it exists. If `jssecacerts` does not exist, then `cacerts` is used. The default location of `cacerts` is `jre\lib\security\`.	String	No

Description

The full path of the Java TrustStore containing the server certificate for one-way SSL authentication.

See also the property SSLTrustStorePwd on page 42.

> ✏ **Note:**
>
> The Simba Impala JDBC Driver accepts TrustStores and KeyStores for one-way
> SSL authentication. See also the property SSLKeyStore on page 40.

SSLTrustStorePwd

Default Value	Data Type	Required
None	String	Yes, if using a TrustStore.

Description

The password for accessing the Java TrustStore that you specified using the property
SSLTrustStore on page 41.

StripCatalogName

Default Value	Data Type	Required
1	Integer	No

Description

This property specifies whether the driver removes catalog names from query
statements if translation fails or if the `UseNativeQuery` property is set to `1`.

- `1`: If query translation fails or if the `UseNativeQuery` property is set to `1`, then
  the driver removes catalog names from the query statement.
- `0`: The driver does not remove catalog names from query statements.

SupportTimeOnlyTimestamp

Default Value	Data Type	Required
1	Integer	No

Description

This property specifies whether the driver supports TIMESTAMP data that only contains a time value.

- `1`: The driver supports TIMESTAMP data that only contains a time value.
- `0`: The driver returns an error when working with TIMESTAMP data that only contains a time value.

UID

Default Value	Data Type	Required
`anonymous`	String	Yes, if `AuthMech=3`.

Description

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

UseNativeQuery

Default Value	Data Type	Required
0	Integer	No

Description

This property specifies whether the driver transforms the queries emitted by applications.

- `1`: The driver does not transform the queries emitted by applications, so the native query is used.
- `0`: The driver transforms the queries emitted by applications and converts them into an equivalent form in Impala SQL.

> ✎ **Note:**
>
> If the application is Impala-aware and already emits Impala SQL, then enable this option to avoid the extra overhead of query transformation.

UseSasl

Default Value	Data Type	Required
1	Integer	No

Description

This property indicates if SASL is used in conjunction with the User Name and Password Authentication Mechanism (`AuthMech=3`).

`0`: No SASL authentication is used. User credentials are still passed to the server for services such as Sentry.

`1`: SASL authentication is used.

Contact Us

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

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

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

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

Third-Party Trademarks

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Cloudera Impala, Cloudera, and Impala are trademarks of Cloudera, Inc. or its subsidiaries in Canada, the United States and/or other countries.

All other trademarks are trademarks of their respective owners.

Third-Party Licenses

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

Simple Logging Façade for Java (SLF4J) License

Copyright © 2004-2015 QOS.ch

All rights reserved.

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

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

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

Apache License, Version 2.0

The following notice is included in compliance with the Apache License, Version 2.0 and is applicable to all software licensed under the Apache License, Version 2.0.

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the

NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or

rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This product includes software that is licensed under the Apache License, Version 2.0 (listed below):

Apache Commons
Copyright © 2001-2015 The Apache Software Foundation

Apache Commons Codec
Copyright © 2002-2014 The Apache Software Foundation

Apache Hadoop Common
Copyright © 2014 The Apache Software Foundation

Apache Hive
Copyright © 2008-2015 The Apache Software Foundation

Apache HttpComponents Client
Copyright © 1999-2012 The Apache Software Foundation

Apache HttpComponents Core
Copyright © 1999-2012 The Apache Software Foundation

Apache Logging Services
Copyright © 1999-2012 The Apache Software Foundation

Apache Thrift
Copyright © 2006-2010 The Apache Software Foundation

Apache ZooKeeper
Copyright © 2010 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.