Simba Apache Hive JDBC Driver with SQL Connector

Installation and Configuration Guide

Simba Technologies Inc.

Version 1.0.36

March 24, 2016

Trademarks

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 Apache Hive JDBC Driver with SQL Connector Installation and Configuration Guide* explains how to install and configure the Simba Apache Hive 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 Apache Hive JDBC Driver with SQL Connector.

Knowledge Prerequisites

To use the Simba Apache Hive JDBC Driver with SQL Connector, the following knowledge is helpful:

- Familiarity with the platform on which you are using the Simba Apache Hive JDBC Driver with SQL Connector
- Ability to use the data store to which the Simba Apache Hive JDBC Driver with SQL Connector 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.

Underline is not used.

✏ The pencil icon indicates a short note appended to a paragraph.

★ The star icon indicates an important comment related to the preceding paragraph.

Table of Contents

Introduction

The Simba Apache Hive JDBC Driver with SQL Connector 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 Features on page 27.

The Simba Apache Hive JDBC Driver with SQL Connector 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 Hive 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 Apache Hive JDBC Driver with SQL Connector 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 Apache Hive versions 0.11 through 1.1.

Simba Apache Hive JDBC Driver with SQL Connector Files

The Simba Apache Hive JDBC Driver with SQL Connector is delivered in the following ZIP archives, where *[Version]* is the version number of the driver:

- `Simba_HiveJDBC3_[Version].zip`
- `Simba_HiveJDBC4_[Version].zip`
- `Simba_HiveJDBC41_[Version].zip`

Each archive contains the driver supporting the JDBC API version indicated in the archive name.

The archives contain the following file and folder structure, where *[LibVersion]* is the version number of the library and *[APIVersion]* is the JDBC API version that the driver supports:

- `HiveJDBC[APIVersion]`
    - `hive_metastore.jar`
    - `hive_service.jar`
    - `HiveJDBC[APIVersion].jar`
    - `libfb303-[LibVersion].jar`
    - `libthrift-[LibVersion].jar`
    - `log4j-[LibVersion].jar`
    - `ql.jar`
    - `Simba JDBC Driver for Hive Install Guide.pdf`
    - `slf4j-api-[LibVersion].jar`
    - `slf4j-log4j12-[LibVersion].jar`
    - `TCLIServiceClient.jar`
    - `zookeeper-[LibVersion].jar`

Using the Simba Apache Hive JDBC Driver with SQL Connector

Before you can use the Simba Apache Hive JDBC Driver with SQL Connector, you must place the `SimbaApacheHiveJDBCDriver.lic` file in the same directory as the `HiveJDBC3.jar`, `HiveJDBC4.jar`, or `HiveJDBC41.jar` file.

To access a Hive data store using the Simba Apache Hive JDBC Driver with SQL Connector, you need to configure the following:

- The class path
- The Driver or DataSource class
- The connection URL for the driver

★ The Simba Apache Hive JDBC Driver with SQL Connector provides read-only access to Hive data.

Setting the Class Path

To use the Simba Apache Hive JDBC Driver with SQL Connector, you must set the class path to include all the JAR files from the ZIP archive containing the driver that you are using.

The class path 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: http://docs.oracle.com/javase/7/docs/technotes/tools/windows/classpath.html.

Initializing the Driver Class

Before connecting to the data store, you must initialize the appropriate class for the Hive server and your application.

The following is a list of the classes used to connect the Simba Apache Hive JDBC Driver with SQL Connector to Hive Server 1 and Hive Server 2 instances. 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.hive.jdbc3.HS1Driver`
- `com.simba.hive.jdbc3.HS2Driver`
- `com.simba.hive.jdbc3.HS1DataSource`
- `com.simba.hive.jdbc3.HS2DataSource`

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

- `com.simba.hive.jdbc4.HS1Driver`
- `com.simba.hive.jdbc4.HS2Driver`
- `com.simba.hive.jdbc4.HS1DataSource`
- `com.simba.hive.jdbc4.HS2DataSource`

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

- `com.simba.hive.jdbc41.HS1Driver`
- `com.simba.hive.jdbc41.HS2Driver`
- `com.simba.hive.jdbc41.HS1DataSource`
- `com.simba.hive.jdbc41.HS2DataSource`

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

```
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.hive.jdbc4.HS1DataSource
   ();
   ds.setURL(CONNECTION_URL);

   connection = ds.getConnection();

   return connection;

}
```

Building the Connection URL

Use the connection URL to supply connection information to the data source that you are accessing. The following is the format of the connection URL for the Simba Apache Hive JDBC Driver with SQL Connector, where *[Subprotocol]* is **hive** if you are connecting to a Hive Server 1 instance or **hive2** if you are connecting to a Hive Server 2 instance, *[Host]* is the DNS or IP address of the Hive server, and *[Port]* is the number of the TCP port that the server uses to listen for client requests:

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

> By default, Hive uses port 10000.

By default, the driver uses the schema named **default** and authenticates the connection using the user name **anonymous**.

You can specify optional settings such as the number of 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 30.

> If you specify a property that is not supported by the driver, then the driver attempts to apply the property as a Hive server-side property for the client session. For more information, see Configuring Server-Side Properties on page 24.

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

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

For example, to connect to port 11000 on a Hive Server 2 instance 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:hive2://localhost:11000/default2;AuthMech=3;
UID=simba;PWD=simba
```

> Be aware of the following:
>
> - Properties are case-sensitive.
> - Do not duplicate properties in the connection URL.

> Note the following:
>
> - If you specify a schema in the connection URL, 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, run the `show databases` command at the Hive command prompt.
> - If you set the `transportMode` property to `http`, then the port number specified in the connection URL corresponds to the HTTP port rather than the TCP port. By default, Hive servers use 10001 as the HTTP port number.

Configuring Authentication

The Simba Apache Hive JDBC Driver with SQL Connector 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 Hive by specifying the relevant properties in the connection URL.

For information about selecting an appropriate authentication mechanism when using the Simba Apache Hive JDBC Driver with SQL Connector, see Authentication Options on page 14.

For information about the properties you can use in the connection URL, see Driver Configuration Options on page 30.

> In addition to authentication, you can configure the driver to connect over SSL. For more information, see Configuring SSL on page 23.

Using No Authentication

> When connecting to a Hive server of type Hive Server 1, you must use No Authentication.

To configure a connection without authentication:

1. Set the `AuthMech` property to `0`.
2. Set the `transportMode` property to `binary`.

For example:

```
jdbc:hive2://localhost:10000;AuthMech=0;
transportMode=binary;
```

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/.

> ✎ Note the following:
>
> - This authentication mechanism is available only for Hive Server 2.
> - When you use this authentication mechanism, SASL is the only Thrift transport protocol that is supported. The driver uses SASL by default, so you do not need to set the `transportMode` property.

To configure 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 Hive server is not the default, then set the `KrbRealm` property to the realm of the Hive server.
3. Set the `KrbHostFQDN` property to the fully qualified domain name of the Hive server host.
4. Set the `KrbServiceName` property to the service name of the Hive server.

For example:

```
jdbc:hive2://localhost:10000;AuthMech=1;
KrbRealm=EXAMPLE.COM;KrbHostFQDN=hs2.example.com;
KrbServiceName=hive
```

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. Most default configurations of Hive Server 2 require User Name authentication.

To configure User Name authentication:

1. Set the `AuthMech` property to `2`.
2. Set the `transportMode` property to `sasl`.
3. Set the `UID` property to an appropriate user name for accessing the Hive server.

For example:

```
jdbc:hive2://localhost:10000;AuthMech=2;
transportMode=sasl;UID=hs2
```

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.

To configure User Name And Password authentication:

1. Set the `AuthMech` property to `3`.
2. Set the `transportMode` property to the transport protocol that you want to use in the Thrift layer.
3. If you set the `transportMode` property to `http`, then set the `httpPath` property to the partial URL corresponding to the Hive server. Otherwise, do not set the `httpPath` property.
4. Set the `UID` property to an appropriate user name for accessing the Hive server.
5. Set the `PWD` property to the password corresponding to the user name you provided.

For example:

```
jdbc:hive2://localhost:10001;AuthMech=3;
transportMode=http;httpPath=cliservice;UID=hs2;
PWD=simba
```

Authentication Options

Hive Server 1 does not support authentication. You must configure the driver to use No Authentication.

Hive Server 2 supports the following authentication mechanisms:

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

Most default configurations of Hive Server 2 require User Name authentication. If you are unable to connect to your Hive server using User Name authentication, then verify the authentication mechanism configured for your Hive server by examining the `hive-site.xml` file. Examine the following properties to determine which authentication mechanism your server is set to use:

- `hive.server2.authentication`: This property sets the authentication mode for Hive Server 2. The following values are available:
  - `NOSASL` disables the Simple Authentication and Security Layer (SASL).
  - `KERBEROS` enables Kerberos authentication.
  - `NONE` enables plain SASL transport. NONE is the default value.
  - `PLAINSASL` enables user name and password authentication using a cleartext password mechanism.

- `hive.server2.enable.doAs`: If this property is set to the default value of `TRUE`, then Hive processes queries as the user who submitted the query. If this property is set to `FALSE`, then queries are run as the user that runs the `hiveserver2` process.

The following table lists the authentication mechanisms to configure for the driver based on the settings in the `hive-site.xml` file.

hive.server2.authentication	hive.server2.enable.doAs	Driver Authentication Mechanism
`NOSASL`	`FALSE`	No Authentication
`KERBEROS`	`TRUE` or `FALSE`	Kerberos
`NONE`	`TRUE` or `FALSE`	User Name
`LDAP`	`TRUE` or `FALSE`	User Name And Password

> It is an error to set `hive.server2.authentication` to `NOSASL` and `hive.server2.enable.doAs` to `true`. This configuration will not prevent the service from starting up, but results in an unusable service.

For more information about authentication mechanisms, refer to the documentation for your Hadoop / Hive distribution. See also "Running Hadoop in Secure Mode" in the Apache Hadoop documentation: http://hadoop.apache.org/docs/r0.23.7/hadoop-project-dist/hadoop-common/ClusterSetup.html#Running_Hadoop_in_Secure_Mode.

Using No Authentication

When `hive.server2.authentication` is set to `NOSASL`, you must configure your connection to use No Authentication.

Using Kerberos

When connecting to a Hive server of type Hive Server 2 and `hive.server2.authentication` is set to `KERBEROS`, you must configure your connection to use Kerberos authentication.

Using User Name

When connecting to a Hive server of type Hive Server 2 and `hive.server2.authentication` is set to `NONE`, you must configure your connection to use User Name authentication. Validation of the credentials that you include depends on `hive.server2.enable.doAs`:

- If `hive.server2.enable.doAs` is set to `TRUE`, then the user name in the driver configuration must be an existing operating system user on the host that is running Hive Server 2.
- If `hive.server2.enable.doAs` is set to `FALSE`, then the user name in the driver configuration is ignored.

If no user name is specified in the driver configuration, then the driver defaults to using **hive** as the user name.

Using User Name And Password

When connecting to a Hive server of type Hive Server 2 and the server is configured to use the SASL-PLAIN authentication mechanism with a user name and a password, you must configure your connection to use User Name And Password authentication.

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.

   > 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 Hive Server

To authenticate to the Hive 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 30.

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 Hive Server

To authenticate to the Hive 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 30.

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:hive2://192.168.1.1:10000";
/**
* 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.

★ 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")
```

Configuring SSL

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

SSL connections require a KeyStore and a TrustStore. You can create a TrustStore and configure the driver to use it, or allow the driver to use one of the default TrustStores. If you do not configure the driver to use a specific TrustStore, then the driver uses the Java TrustStore **jssecacerts**. If **jssecacerts** is not available, then the driver uses **cacerts** instead.

To configure SSL:

1. Create a KeyStore and configure the driver to use it:
   a. Create a KeyStore containing your signed, trusted SSL certificate.
   b. Set the `SSLKeyStore` property to the full path of the KeyStore, including the file name.
   c. Set the `SSLKeyStorePwd` property to the password for the KeyStore.
2. Optionally, create a TrustStore and configure the driver to use it:
   a. Create a TrustStore containing your signed, trusted SSL certificate.
   b. Set the `SSLTrustStore` property to the full path of the TrustStore, including the file name.
   c. Set the `SSLTrustStorePwd` property to the password for the TrustStore.
3. Set the `SSL` property to `1`.
4. Optionally, to disable host name verification, set the `AllowAllHostNames` property to `1`.

   > ★ For security reasons, it is strongly recommended that you do not disable host name verification.

5. Optionally, to allow the SSL certificate used by the server to be self-signed, set the `AllowSelfSignedCerts` property to `1`.
6. Optionally, to allow the common name of a CA-issued certificate to not match the host name of the Hive server, set the `CAIssuedCertNamesMismatch` property to `1`.

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

For example:

```
jdbc:hive2://localhost:10000;AuthMech=3;SSL=1;
SSLKeyStore=C:\\Users\\bsmith\\Desktop\\keystore.jks;
SSLKeyStorePwd=*****;UID=hs2;PWD=*****
```

> ✎ For more information about the connection properties used in SSL connections, see Driver Configuration Options on page 30

Configuring Server-Side Properties

You can use the driver to apply configuration properties to the Hive server by setting the properties in the connection URL.

For example, to set the `mapreduce.job.queuename` property to `myQueue`, you would use a connection URL such as the following:

```
jdbc:hive://localhost:18000/default2;AuthMech=3;
UID=simba;PWD=simba;mapreduce.job.queuename=myQueue
```

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

Configuring Logging

To help troubleshoot issues, you can enable logging in the driver.

> ★ Only enable logging long enough to capture an issue. Logging decreases performance and can consume a large quantity of disk space.

Set the `LogLevel` key in your connection URL to enable logging and specify the amount of detail included in log files. The following table lists the logging levels provided by the Simba Apache Hive JDBC Driver with SQL Connector, 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 potentially harmful situations.
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.

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

   ```
   jdbc:hive://localhost:11000;LogLevel=3;LogPath=C:\temp
   ```
3. To make sure that the new settings take effect, restart your JDBC application and reconnect to the server.

The Simba Apache Hive JDBC Driver with SQL Connector produces the following log files in the location specified in the `LogPath` property:

- A `HiveJDBC_driver.log` file that logs driver activity that is not specific to a connection.
- A `HiveJDBC_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 Apache Hive JDBC Driver with SQL Connector:

- SQL Query versus HiveQL Query on page 27
- Data Types on page 27
- Catalog and Schema Support on page 28

SQL Query versus HiveQL Query

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

Data Types

The Simba Apache Hive JDBC Driver with SQL Connector supports many common data formats, converting between Hive, SQL, and Java data types.

The following table lists the supported data type mappings.

Hive Type	SQL Type	Java Type
BIGINT	BIGINT	java.math.BigInteger
BINARY	VARBINARY	byte[]
BOOLEAN	BOOLEAN	Boolean
CHAR (Available only in Hive 0.13.0 or later)	CHAR	String
DATE	DATE	java.sql.Date
DECIMAL (In Hive 0.13 and later, you can specify scale and precision when creating tables using the DECIMAL data type.)	DECIMAL	java.math.BigDecimal

Hive Type	SQL Type	Java Type
DOUBLE	DOUBLE	Double
FLOAT	REAL	Float
INT	INTEGER	Long
SMALLINT	SMALLINT	Integer
TIMESTAMP	TIMESTAMP	java.sql.Timestamp
TINYINT	TINYINT	Short
VARCHAR (Available only in Hive 0.12.0 or later)	VARCHAR	String

The aggregate types (ARRAY, MAP, STRUCT, and UNIONTYPE) are not yet supported. Columns of aggregate types are treated as VARCHAR columns in SQL and STRING columns in Java.

Catalog and Schema Support

The Simba Apache Hive JDBC Driver with SQL Connector supports both catalogs and schemas to make it easy for the driver to work with various JDBC 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 JDBC schema to the Hive schema/database.

> By default, Hive catalogs are treated as schemas in the driver as a restriction for filtering. To configure the driver to treat Hive catalogs as catalogs and Hive schemas as schemas, set the `CatalogSchemaSwitch` connection property to `0`.

Contact Us

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

Technical Support is available Monday to Friday from 8 a.m. to 6 p.m. Pacific Time.

> ★ 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:

- **E-mail**: support@simba.com
- **Web site**: www.simba.com
- **Telephone**: (604) 633-0008 Extension 3
- **Fax**: (604) 633-0004

You can also follow us on Twitter @SimbaTech

Driver Configuration Options

Driver Configuration Options lists and describes the properties that you can use to configure the behavior of the Simba Apache Hive JDBC Driver with SQL Connector.

You can set configuration properties using the connection URL. For more information, see Building the Connection URL on page 11.

AllowAllHostNames

Default Value	Data Type	Required
`0`	Integer	No

Description

This property specifies whether host name verification is enabled for SSL connections.

- `0`: Host name verification is enabled, so the driver requires the host name specified in the SSL certificate to match the domain of the URL being requested.
- `1`: Host name verification is disabled, so the driver accepts all host names.

> ★ For security reasons, it is strongly recommended that you do not disable host name verification.
>
> ✎ This property is applicable only when SSL connections are enabled.

AllowSelfSignedCerts

Default Value	Data Type	Required
`0`	Integer	No

Description

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

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

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

AuthMech

Default Value	Data Type	Required
Depends on the `transportMode` setting. For more information, see transportMode on page 38.	Integer	No

Description

The authentication mechanism to use. Set the value to one of the following numbers:

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

CAIssuedCertNamesMismatch

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 Hive server.

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

This property is applicable only when SSL connections are enabled.

CatalogSchemaSwitch

Default Value	Data Type	Required
`1`	Integer	No

Description

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

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

DecimalColumnScale

Default Value	Data Type	Required
`10`	Integer	No

Description

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

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 Hive 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 Hive to a user that is different than the authenticated user for the connection.

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

httpPath

Default Value	Data Type	Required
`None`	String	Yes, if `transportMode=http`.

Description

The partial URL corresponding to the Hive server.

The driver forms the HTTP address to connect to by appending the `httpPath` value to the host and port specified in the connection URL. For example, to connect to the HTTP address `http://localhost:10002/cliservice`, you would use the following connection URL:

```
jdbc:hive2://localhost:10002;AuthMech=3;
transportMode=http;httpPath=cliservice;UID=hs2;PWD=simba;
```

✎ By default, Hive servers use `cliservice` as the partial URL.

KrbHostFQDN

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

Description

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

KrbRealm

Default Value	Data Type	Required
Depends on Kerberos configuration.	String	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.

KrbServiceName

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

Description

The Kerberos service principal name of the Hive 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.

> ★ 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 potentially harmful situations.
- `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 `HiveJDBC_driver.log` file that logs driver activity that is not specific to a connection.

- A `HiveJDBC_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.

PreparedMetaLimitZero

Default Value	Data Type	Required
0	Integer	No

Description

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

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

PWD

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

Description

The password corresponding to the user name that you provided using the property UID on page 39.

★ If you set the `AuthMech` to `3`, the default `PWD` value is not used and you must specify a password.

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 Hive 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 Hive server through an SSL-enabled socket.

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

> ✎ 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	Yes, if `SSL=1`.

Description

The full path and file name of the Java KeyStore containing an SSL certificate to use during authentication.

See also the property SSLKeyStorePwd on page 37.

SSLKeyStorePwd

Default Value	Data Type	Required
None	Integer	Yes, if `SSL=1`.

Description

The password for accessing the Java KeyStore that you specified using the property SSLKeyStore on page 37.

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 and file name of the Java TrustStore containing an SSL certificate to use during authentication.

See also the property SSLTrustStorePwd on page 38.

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 37.

transportMode

Default Value	Data Type	Required
`sasl`	String	No

Description

The transport protocol to use in the Thrift layer.

- `binary`: The driver uses the Binary transport protocol.

  When connecting to a Hive Server 1 instance, you must use this setting. If you use this setting but do not specify the `AuthMech` property, then the driver uses `AuthMech=0` by default. This setting is valid only when the `AuthMech` property is set to `0` or `3`.
- `sasl`: The driver uses the SASL transport protocol.

  If you use this setting but do not specify the `AuthMech` property, then the driver uses `AuthMech=2` by default. This setting is valid only when the `AuthMech` property is set to `1`, `2`, or `3`.
- `http`: The driver uses the HTTP transport protocol.

  When connecting to Hive through the Apache Knox Gateway, you must use this setting. If you use this setting but do not specify the `AuthMech` property, then the driver uses `AuthMech=3` by default. This setting is valid only when the `AuthMech` property is set to `3`.

  If you set this property to `http`, then the port number in the connection URL corresponds to the HTTP port rather than the TCP port, and you must specify the `httpPath` property. For more information, see httpPath on page 33.

UID

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

Description

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

> ★ If you set the `AuthMech` to `3`, the default `UID` value is not used and you must specify a user name.

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 HiveQL.

> ✎ If the application is Hive-aware and already emits HiveQL, then enable this option to avoid the extra overhead of query transformation.

zk

Default Value	Data Type	Required
None	String	No

Description

The connection string to one or more ZooKeeper quorums, written in the following format where *[ZK_IP]* is the IP address, *[ZK_Port]* is the port number, and *[ZK_Namespace]* is the namespace:

```
[ZK_IP]:[ZK_Port]/[ZK_Namespace]
```

For example:

```
jdbc:hive2://zk=192.168.0.1:2181/hiveserver2
```

Use this option to enable the Dynamic Service Discovery feature, which allows you to connect to Hive servers that are registered against a ZooKeeper service by connecting to the ZooKeeper service.

You can specify multiple quorums in a comma-separated list. If connection to a quorum fails, the driver will attempt to connect to the next quorum in the list.

Third-Party Trademarks

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

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.

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

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