

Cloudera JDBC Driver for Apache
Hive

Version 2.5.17



Important Notice

© 2010-2016 Cloudera, Inc. All rights reserved.

Cloudera, the Cloudera logo, and any other product or service names or slogans contained in this document, except as otherwise disclaimed, are trademarks of Cloudera and its suppliers or licensors, and may not be copied, imitated or used, in whole or in part, without the prior written permission of Cloudera or the applicable trademark holder.

Hadoop and the Hadoop elephant logo are trademarks of the Apache Software Foundation. All other trademarks, registered trademarks, product names and company names or logos mentioned in this document are the property of their respective owners. Reference to any products, services, processes or other information, by trade name, trademark, manufacturer, supplier or otherwise does not constitute or imply endorsement, sponsorship or recommendation thereof by us.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Cloudera.

Cloudera may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Cloudera, the furnishing of this document does not give you any license to these patents, trademarks copyrights, or other intellectual property.

The information in this document is subject to change without notice. Cloudera shall not be liable for any damages resulting from technical errors or omissions which may be present in this document, or from use of this document.

Cloudera, Inc.
1001 Page Mill Road, Building 2
Palo Alto, CA 94304-1008
info@cloudera.com
US: 1-888-789-1488
Intl: 1-650-843-0595
www.cloudera.com

Release Information

Version: 2.5.17

Date: September 29, 2016

Table of Contents

ABOUT THE CLUDERA JDBC DRIVER FOR APACHE HIVE	5
SYSTEM REQUIREMENTS	6
CLUDERA JDBC DRIVER FOR APACHE HIVE FILES	7
INSTALLING AND USING THE CLUDERA JDBC DRIVER FOR APACHE HIVE	8
SETTING THE CLASS PATH	8
REGISTERING THE DRIVER CLASS	8
BUILDING THE CONNECTION URL	10
CONFIGURING AUTHENTICATION	12
USING NO AUTHENTICATION	12
USING KERBEROS	12
USING USER NAME	13
USING USER NAME AND PASSWORD	14
AUTHENTICATION MECHANISMS	14
CONFIGURING KERBEROS AUTHENTICATION FOR WINDOWS	16
CONFIGURING SSL	22
CONFIGURING SERVER-SIDE PROPERTIES	24
CONFIGURING LOGGING	25
FEATURES	27
SQL QUERY VERSUS HIVEQL QUERY	27
DATA TYPES	27
CATALOG AND SCHEMA SUPPORT	28
INTERFACES AND SUPPORTED METHODS	28
CONTACT Us	84
DRIVER CONFIGURATION OPTIONS	85
ALLOWALLHOSTNAMES	85
ALLOWSELF-SIGNEDCERT	85
AUTHMECH	86
CAISSUEDCERTNAMESMISMATCH	86
CATALOGSCHEMASWITCH	86
DECIMALCOLUMNSCALE	87
DEFAULTSTRINGCOLUMNLENGTH	87
DELEGATIONUID	87
HTTPPATH	88
KRBHOSTFQDN	88
KRBREALM	88

KRBSERVICENAME	89
LOGLEVEL	89
LOGPATH	90
PREPAREDMETALIMITZERO	90
PWD	90
ROWSFETCHEDPERBLOCK	91
SOCKETTIMEOUT	91
SSL	91
SSLKEYSTORE	92
SSLKEYSTOREPWD	92
SSLTRUSTSTORE	92
SSLTRUSTSTOREPWD	92
TRANSPORTMODE	93
UID	93
USENATIVEQUERY	94
ZK	94

About the Cloudera JDBC Driver for Apache Hive

The Cloudera JDBC Driver for Apache Hive 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 Cloudera JDBC Driver for Apache Hive complies with the JDBC 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 Cloudera JDBC Driver for Apache Hive 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
4.0	6.0 or later
4.1	7.0 or later

The driver supports Apache Hive versions 0.11 through 1.1.

Cloudera JDBC Driver for Apache Hive Files

The Cloudera JDBC Driver for Apache Hive is delivered in the following ZIP archives, where *[Version]* is the version number of the driver:

- HiveJDBC4_*[Version]*.zip
- 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
 - slf4j-api-*[LibVersion]*.jar
 - slf4j-log4j12-*[LibVersion]*.jar
 - TCLIServiceClient.jar
 - zookeeper-*[LibVersion]*.jar

Installing and Using the Cloudera JDBC Driver for Apache Hive

To install the Cloudera JDBC Driver for Apache Hive on your machine, extract the files from the appropriate ZIP archive to the directory of your choice.

To access a Hive data store using the Cloudera JDBC Driver for Apache Hive, you need to configure the following:

- The class path (see "Setting the Class Path" on page 8)
- The Driver or DataSource class (see "Registering the Driver Class" on page 8)
- The connection URL for the driver (see "Building the Connection URL" on page 10)

Important:

The Cloudera JDBC Driver for Apache Hive is a forward-only, read-only driver with no transaction support. Because the driver does not support transactions, auto-commit is always set to **true**.

Setting the Class Path

To use the Cloudera JDBC Driver for Apache Hive, 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:

- 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 Cloudera JDBC Driver for Apache Hive 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 4.0, classes with the following FQCNs are available:

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

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

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

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.cloudera.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 Cloudera JDBC Driver for Apache Hive, 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]
```

Note:

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 85.

Note:

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=cloudera;PWD=cloudera
```

Important:

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

Note:

- 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 Cloudera JDBC Driver for Apache Hive 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 Cloudera JDBC Driver for Apache Hive, see "Authentication Mechanisms" on page 14.

For information about the properties you can use in the connection URL, see "Driver Configuration Options" on page 85.

Note:

In addition to authentication, you can configure the driver to connect over SSL. For more information, see "Configuring SSL" on page 22.

Using No Authentication

Note:

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:

- 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.

Note:

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.

Note:

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=cloudera;
```

Authentication Mechanisms

Hive Server 1 does not support authentication. You must configure the driver to use No Authentication (see "Using No Authentication" on page 12).

Hive Server 2 supports the following authentication mechanisms:

- No Authentication (see "Using No Authentication" on page 12)
- Kerberos (see "Using Kerberos" on page 12)
- User Name (see "Using User Name" on page 13)
- User Name And Password (see "Using User Name And Password" on page 14)

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.

<code>hive.server2.authentication</code>	<code>hive.server2.enable.doAs</code>	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

Note:

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>.

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 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 85

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="cloudera@CLUDERA"  
  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 85.

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;
```

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.

Important:

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.

Note:

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=*****
```

Note:

For more information about the connection properties used in SSL connections, see "Driver Configuration Options" on page 85

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=cloudera;PWD=cloudera;mapreduce.job.queueName=myQueue
```

Note:

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.

Important:

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 Cloudera JDBC Driver for Apache Hive, 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: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 Cloudera JDBC Driver for Apache Hive 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.

Configuring Logging

- 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 Cloudera JDBC Driver for Apache Hive:

- "SQL Query versus HiveQL Query" on page 27
- "Data Types" on page 27
- "Catalog and Schema Support" on page 28
- "Interfaces and Supported Methods" 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 Cloudera JDBC Driver for Apache Hive 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
DOUBLE	DOUBLE	Double
FLOAT	REAL	Float

Hive Type	SQL Type	Java Type
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 Cloudera JDBC Driver for Apache Hive 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.

Note:

Setting the `CatalogSchemaSwitch` connection property to 1 will cause Hive catalogs to be treated as schemas in the driver as a restriction for filtering.

Interfaces and Supported Methods

The Cloudera JDBC Driver for Apache Hive implements the following JDBC interfaces:

- "CallableStatement" on page 29
- "Connection" on page 38
- "DatabaseMetaData" on page 43
- "DataSource" on page 56
- "Driver" on page 57
- "ParameterMetaData" on page 57
- "PooledConnection" on page 58
- "PreparedStatement" on page 59
- "ResultSet" on page 64
- "ResultSetMetaData" on page 79
- "Statement" on page 80

However, the driver does not support every method from these interfaces. For information about whether a specific method is supported by the driver and which version of the JDBC API is the earliest version that supports the method, refer to the following sections.

The driver does not support the following JDBC features:

- Array
- Blob
- Clob
- Ref
- Savepoint
- SQLData
- SQLInput
- SQLOutput
- Struct

CallableStatement

The `CallableStatement` interface extends the `PreparedStatement` interface.

The following table lists the methods that belong to the `CallableStatement` interface, and describes whether each method is supported by the Cloudera JDBC Driver for Apache Hive and which version of the JDBC API is the earliest version that supports the method.

For detailed information about each method in the `CallableStatement` interface, see the Java API documentation:

<http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/CallableStatement.html>.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>Array getArray(int i)</code>	3.0	No	
<code>Array getArray(String parameterName)</code>	3.0	No	
<code>BigDecimal getBigDecimal(int parameterIndex)</code>	3.0	Yes	
<code>BigDecimal getBigDecimal(int parameterIndex, int scale)</code>	3.0	Yes	Deprecated.
<code>BigDecimal getBigDecimal(String parameterName)</code>	3.0	Yes	
<code>Blob getBlob(int i)</code>	3.0	No	
<code>Blob getBlob(String parameterName)</code>	3.0	No	
<code>boolean getBoolean(int parameterIndex)</code>	3.0	Yes	

Features

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>boolean getBoolean(String parameterName)</code>	3.0	Yes	
<code>byte getByte(int parameterIndex)</code>	3.0	Yes	
<code>byte getByte(String parameterName)</code>	3.0	Yes	
<code>byte[] getBytes(int parameterIndex)</code>	3.0	Yes	
<code>byte[] getBytes(String parameterName)</code>	3.0	Yes	
<code>Clob getClob(int i)</code>	3.0	No	
<code>Clob getClob(String parameterName)</code>	3.0	No	
<code>Date getDate(int parameterIndex)</code>	3.0	Yes	
<code>Date getDate(int parameterIndex, Calendar cal)</code>	3.0	Yes	
<code>Date getDate(String parameterName)</code>	3.0	Yes	
<code>Date getDate(String parameterName, Calendar cal)</code>	3.0	Yes	
<code>double getDouble(int parameterIndex)</code>	3.0	Yes	
<code>double getDouble(String parameterName)</code>	3.0	Yes	
<code>float getFloat(int parameterIndex)</code>	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>float getFloat(String parameterName)</code>	3.0	Yes	
<code>int getInt(int parameterIndex)</code>	3.0	Yes	
<code>int getInt(String parameterName)</code>	3.0	Yes	
<code>long getLong(int parameterIndex)</code>	3.0	Yes	
<code>long getLong(String parameterName)</code>	3.0	Yes	
<code>Reader getNCharacterStream(int parameterIndex)</code>	4.0	No	
<code>Reader getNCharacterStream (String parameterName)</code>	4.0	No	
<code>NClob getNClob(int parameterIndex)</code>	4.0	No	
<code>NClob getNClob (String parameterName)</code>	4.0	No	
<code>String getNString(int parameterIndex)</code>	4.0	No	
<code>String getNString (String parameterName)</code>	4.0	No	
<code>Object getObject(int parameterIndex)</code>	3.0	Yes	
<code><T> T getObject(int parameterIndex, Class<T> type)</code>	4.1	No	
<code>Object getObject(int i,</code>	3.0	No	

Features

Method	Supported Since JDBC Version	Supported by the Driver	Notes
Map<String,Class<?>> map)			
Object getObject(String parameterName)	3.0	Yes	
<T> T getObject(String parameterName, Class<T> type)	4.1	No	
Object getObject(String parameterName, Map<String,Class<?>> map)	3.0	Yes	
Ref getRef(int i)	3.0	No	
Ref getRef(String parameterName)	3.0	No	
RowId getRowId(int parameterIndex)	4.0	No	
RowId getRowId(String parameterName)	4.0	No	
short getShort(int parameterIndex)	3.0	Yes	
short getShort(String parameterName)	3.0	Yes	
SQLXML getSQLXML(int parameterIndex)	4.0	No	
SQLXML getSQLXML(String parameterName)	4.0	No	
String getString(int parameterIndex)	3.0	Yes	
String getString(String parameterName)	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
Time getTime(int parameterIndex)	3.0	Yes	
Time getTime(int parameterIndex, Calendar cal)	3.0	Yes	
Time getTime(String parameterName)	3.0	Yes	
Time getTime(String parameterName, Calendar cal)	3.0	Yes	
Timestamp getTimestamp(int parameterIndex)	3.0	Yes	
Timestamp getTimestamp(int parameterIndex, Calendar cal)	3.0	Yes	
Timestamp getTimestamp(String parameterName)	3.0	Yes	
Timestamp getTimestamp(String parameterName, Calendar cal)	3.0	Yes	
URL getURL(int parameterIndex)	3.0	No	
URL getURL(String parameterName)	3.0	No	
void registerOutParameter(int parameterIndex, int sqlType)	3.0	Yes	
void registerOutParameter(int parameterIndex, int sqlType, int scale)	3.0	Yes	

Features

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>void registerOutParameter(int paramIndex, int sqlType, String typeName)</code>	3.0	Yes	
<code>void registerOutParameter (String parameterName, int sqlType)</code>	3.0	Yes	
<code>void registerOutParameter (String parameterName, int sqlType, int scale)</code>	3.0	Yes	
<code>void registerOutParameter (String parameterName, int sqlType, String typeName)</code>	3.0	Yes	
<code>void setAsciiStream (String parameterName, InputStream x)</code>	4.0	Yes	
<code>void setAsciiStream (String parameterName, InputStream x, int length)</code>	3.0	Yes	
<code>void setAsciiStream (String parameterName, InputStream x, long length)</code>	4.0	Yes	
<code>void setBigDecimal (String parameterName, BigDecimal x)</code>	3.0	Yes	
<code>void setBinaryStream (String parameterName, InputStream x)</code>	4.0	Yes	
<code>setBinaryStream(String</code>	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>parameterName, InputStream x, int length)</code>			
<code>void setBinaryStream (String parameterName, InputStream x, long length)</code>	4.0	Yes	
<code>void setBlob(String parameterName, Blob x)</code>	4.0	Yes	
<code>void setBlob(String parameterName, InputStream inputStream)</code>	4.0	Yes	
<code>void setBlob(String parameterName, InputStream inputStream, long length)</code>	4.0	Yes	
<code>void setBoolean(String parameterName, boolean x)</code>	3.0	Yes	
<code>void setByte(String parameterName, byte x)</code>	3.0	Yes	
<code>void setBytes(String parameterName, byte[] x)</code>	3.0	Yes	
<code>void setCharacterStream (String parameterName, Reader reader)</code>	4.0	Yes	
<code>void setCharacterStream (String parameterName, Reader reader, int length)</code>	3.0	Yes	
<code>void setCharacterStream (String parameterName, Reader reader, long length)</code>	4.0	Yes	

Features

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>void setClob(String parameterName, Clob x)</code>	4.0	Yes	
<code>void setClob(String parameterName, Reader reader)</code>	4.0	Yes	
<code>void setClob(String parameterName, Reader reader, long length)</code>	4.0	Yes	
<code>void setDate(String parameterName, Date x)</code>	3.0	Yes	
<code>void setDate(String parameterName, Date x, Calendar cal)</code>	3.0	Yes	
<code>void setDouble(String parameterName, double x)</code>	3.0	Yes	
<code>void setFloat(String parameterName, float x)</code>	3.0	Yes	
<code>void setInt(String parameterName, int x)</code>	3.0	Yes	
<code>void setLong(String parameterName, long x)</code>	3.0	Yes	
<code>void setNCharacterStream(String parameterName, Reader value)</code>	4.0	Yes	
<code>void setNCharacterStream(String parameterName, Reader value, long length)</code>	4.0	Yes	
<code>void setNClob(String parameterName, NClob value)</code>	4.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>void setNClob(String parameterName, Reader reader)</code>	4.0	Yes	
<code>void setNClob(String parameterName, Reader reader, long length)</code>	4.0	Yes	
<code>void setNString(String parameterName, String value)</code>	4.0	Yes	
<code>void setNull(String parameterName, int sqlType)</code>	3.0	Yes	
<code>void setNull(String parameterName, int sqlType, String typeName)</code>	3.0	Yes	
<code>void setObject(String parameterName, Object x)</code>	3.0	Yes	
<code>void setObject(String parameterName, Object x, int targetSqlType)</code>	3.0	Yes	
<code>void setObject(String parameterName, Object x, int targetSqlType, int scale)</code>	3.0	Yes	
<code>void setRowId(String parameterName, RowId x)</code>	4.0	Yes	
<code>void setShort(String parameterName, short x)</code>	3.0	Yes	
<code>void setSQLXML(String parameterName, SQLXML xmlObject)</code>	4.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>void setString(String parameterName, String x)</code>	3.0	Yes	
<code>void setTime(String parameterName, Time x)</code>	3.0	Yes	
<code>void setTime(String parameterName, Time x, Calendar cal)</code>	3.0	Yes	
<code>void setTimestamp(String parameterName, Timestamp x)</code>	3.0	Yes	
<code>void setTimestamp(String parameterName, Timestamp x, Calendar cal)</code>	3.0	Yes	
<code>void setURL(String parameterName, URL val)</code>	3.0	Yes	
<code>boolean wasNull()</code>	3.0	Yes	
<code>boolean isWrapperFor(Class<?> iface)</code>	4.0	Yes	
<code><T> T unwrap(Class<T> iface)</code>	4.0	Yes	

Connection

The following table lists the methods that belong to the `Connection` interface, and describes whether each method is supported by the Cloudera JDBC Driver for Apache Hive and which version of the JDBC API is the earliest version that supports the method.

For detailed information about each method in the `Connection` interface, see the Java API documentation: <http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/Connection.html>.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>void clearWarnings()</code>	3.0	Yes	
<code>void close()</code>	3.0	Yes	
<code>void commit()</code>	3.0	Yes	Auto-commit cannot be set to <code>false</code> because it is hard-coded to <code>true</code> .
Array <code>createArrayOf (String typeName, Object [] elements)</code>	4.0	No	
Blob <code>createBlob()</code>	4.0	No	
Clob <code>createClob()</code>	4.0	No	
NClob <code>createNClob()</code>	4.0	No	
SQLXML <code>createSQLXML()</code>	4.0	No	
Statement <code>createStatement ()</code>	3.0	Yes	
Statement <code>createStatement (int resultSetType, int resultSetConcurrency)</code>	3.0	No	
Statement <code>createStatement (int resultSetType, int resultSetConcurrency, int resultSetHoldability)</code>	3.0	No	
Struct <code>createStruct (String typeName, Object [] attributes)</code>	4.0	No	
<code>boolean getAutoCommit()</code>	3.0	Yes	Hard-coded to <code>true</code> .
<code>String getCatalog()</code>	3.0	Yes	
Properties <code>getClientInfo ()</code>	4.0	Yes	

Features

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>String getClientInfo (String name)</code>	4.0	Yes	
<code>int getHoldability()</code>	3.0	Yes	Hard-coded to <code>CLOSE_CURSORS_AT_COMMIT</code> .
<code>DatabaseMetaData getMetaData()</code>	3.0	Yes	
<code>int getNetworkTimeout()</code>	4.1	No	
<code>String getSchema()</code>	4.1	Yes	The returned schema name does not always match the one used by statements. Statements use the schema name defined in the connection URL.
<code>int getTransactionIsolation()</code>	3.0	Yes	Hard-coded to <code>TRANSACTION_READ_UNCOMMITTED</code> .
<code>Map<String,Class<?>> getTypeMap()</code>	3.0	No	
<code>SQLWarning getWarnings()</code>	3.0	Yes	
<code>boolean isClosed()</code>	3.0	Yes	
<code>boolean isReadOnly()</code>	3.0	Yes	Returns <code>true</code> .
<code>boolean isValid(int timeout)</code>	4.0	Yes	
<code>String nativeSQL(String sql)</code>	3.0	Yes	
<code>CallableStatement prepareCall(String sql)</code>	3.0	No	
<code>CallableStatement prepareCall(String sql,</code>	3.0	No	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>int resultSetType, int resultSetConcurrency)</code>			
<code>CallableStatement prepareCall(String sql, int resultSetType, int resultSetConcurrency, int resultSetHoldability)</code>	3.0	No	
<code>PreparedStatement prepareStatement(String sql)</code>	3.0	Yes	
<code>PreparedStatement prepareStatement(String sql, int autoGeneratedKeys)</code>	3.0	No	
<code>PreparedStatement prepareStatement(String sql, int[] columnIndexes)</code>	3.0	No	
<code>PreparedStatement prepareStatement(String sql, int resultSetType, int resultSetConcurrency)</code>	3.0	No	
<code>PreparedStatement prepareStatement(String sql, int resultSetType, int resultSetConcurrency, int resultSetHoldability)</code>	3.0	No	
<code>PreparedStatement prepareStatement(String sql, String[] columnNames)</code>	3.0	No	
<code>void releaseSavepoint (Savepoint savepoint)</code>	3.0	No	Savepoints are not available because transactions are not supported.
<code>void rollback()</code>	3.0	No	Savepoints are not available

Features

Method	Supported Since JDBC Version	Supported by the Driver	Notes
			because transactions are not supported.
<code>void rollback(Savepoint savepoint)</code>	3.0	No	Savepoints are not available because transactions are not supported.
<code>void setAutoCommit(boolean autoCommit)</code>	3.0	Yes	Ignored because auto-commit is hard-coded to true.
<code>void setCatalog(String catalog)</code>	3.0	Yes	
<code>void setClientInfo(Properties properties)</code>	4.0	Yes	
<code>void setClientInfo(String name, String value)</code>	4.0	Yes	
<code>void setHoldability(int holdability)</code>	3.0	Yes	
<code>void setNetworkTimeout(Executor executor, int milliseconds)</code>	4.1	No	
<code>void setReadOnly(boolean readOnly)</code>	3.0	Yes	
<code>Savepoint setSavepoint()</code>	3.0	No	Savepoints are not available because transactions are not supported.
<code>Savepoint setSavepoint(String name)</code>	3.0	No	Savepoints are not available because transactions are not supported.
<code>void setSchema(String schema)</code>	4.1	Yes	Does not actually change the schema name used by newly created statements; only changes the value

Method	Supported Since JDBC Version	Supported by the Driver	Notes
			returned by <code>getSchema()</code> .
<code>void setTransactionIsolation(int level)</code>	3.0	Yes	
<code>void setTypeMap(Map<String, Class<?>> map)</code>	3.0	No	
<code>boolean isWrapperFor(Class<?> iface)</code>	4.0	Yes	
<code><T> T unwrap(Class<T> iface)</code>	4.0	Yes	

DatabaseMetaData

The following table lists the methods that belong to the `DatabaseMetaData` interface, and describes whether each method is supported by the Cloudera JDBC Driver for Apache Hive and which version of the JDBC API is the earliest version that supports the method.

For detailed information about each method in the `DatabaseMetaData` interface, see the Java API

documentation: <http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/DatabaseMetaData.html>.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>boolean allProceduresAreCallable()</code>	3.0	Yes	Returns <code>true</code> .
<code>boolean allTablesAreSelectable()</code>	3.0	Yes	Returns <code>true</code> .
<code>boolean autoCommitFailureClosesAllResultSets()</code>	4.0	Yes	Returns <code>true</code> .

Features

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>boolean dataDefinitionCausesTransactionCommit()</code>	3.0	Yes	Returns false.
<code>boolean dataDefinitionIgnoredInTransactions()</code>	3.0	Yes	Returns false.
<code>boolean deletesAreDetected(int type)</code>	3.0	Yes	Returns true.
<code>boolean doesMaxRowSizeIncludeBlobs()</code>	3.0	Yes	Returns false.
<code>boolean generatedKeyAlwaysReturned()</code>	4.1	Yes	
<code>ResultSet getAttributes(String catalog, String schemaPattern, String typeNamePattern, String attributeNamePattern)</code>	3.0	Yes	
<code>ResultSet getBestRowIdentifier(String catalog, String schema, String table, int scope, boolean nullable)</code>	3.0	Yes	
<code>ResultSet getCatalogs()</code>	3.0	Yes	
<code>String getCatalogSeparator()</code>	3.0	Yes	
<code>String getCatalogTerm()</code>	3.0	Yes	
<code>ResultSet getClientInfoProperties()</code>	4.0	Yes	
<code>ResultSet getColumnPrivileges(String catalog, String schema, String table, String columnNamePattern)</code>	3.0	Yes	
<code>ResultSet getColumns(String catalog, String schemaPattern, String tableNamePattern, String columnNamePattern)</code>	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
Connection getConnection()	3.0	Yes	
ResultSet getCrossReference(String primaryCatalog, String primarySchema, String primaryTable, String foreignCatalog, String foreignSchema, String foreignTable)	3.0	Yes	
int getDatabaseMajorVersion()	3.0	Yes	
int getDatabaseMinorVersion()	3.0	Yes	
String getDatabaseProductName()	3.0	Yes	Hard-coded to Impala.
String getDatabaseProductVersion()	3.0	Yes	
int getDefaultTransactionIsolation()	3.0	Yes	Hard-coded to TRANSACTION_READ_UNCOMMITTED.
int getDriverMajorVersion()	3.0	Yes	
int getDriverMinorVersion()	3.0	Yes	
String getDriverName()	3.0	Yes	Hard-coded to ImpalaJDBC.
String getDriverVersion()	3.0	Yes	
ResultSet getExportedKeys(String catalog, String schema, String table)	3.0	Yes	
String getExtraNameCharacters()	3.0	Yes	Returns an empty String.
ResultSet getFunctionColumns(String catalog, String schemaPattern,	4.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
String functionNamePattern, String columnNamePattern)			
ResultSet getFunctions(String catalog, String schemaPattern, String functionNamePattern)	4.0	Yes	
String getIdentifierQuoteString()	3.0	Yes	Returns a backquote (`)
ResultSet getImportedKeys(String catalog, String schema, String table)	3.0	Yes	
ResultSet getIndexInfo(String catalog, String schema, String table, boolean unique, boolean approximate)	3.0	Yes	
int getJDBCMajorVersion()	3.0	Yes	
int getJDBCMinorVersion()	3.0	Yes	
int getMaxBinaryLiteralLength()	3.0	Yes	Returns 0.
int getMaxCatalogNameLength()	3.0	Yes	Returns 128.
int getMaxCharLiteralLength()	3.0	Yes	Returns 0.
int getMaxColumnNameLength()	3.0	Yes	Returns 128.
int getMaxColumnsInGroupBy()	3.0	Yes	Returns 0.
int getMaxColumnsInIndex()	3.0	Yes	Returns 0.
int getMaxColumnsInOrderBy()	3.0	Yes	Returns 0.
int getMaxColumnsInSelect()	3.0	Yes	Returns 0.
int getMaxColumnsInTable()	3.0	Yes	Returns 0.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>int getMaxConnections()</code>	3.0	Yes	Returns 0.
<code>int getMaxCursorNameLength()</code>	3.0	Yes	Returns 0.
<code>int getMaxIndexLength()</code>	3.0	Yes	Returns 0.
<code>int getMaxProcedureNameLength()</code>	3.0	Yes	Returns 0.
<code>int getMaxRowSize()</code>	3.0	Yes	Returns 0.
<code>int getMaxSchemaNameLength()</code>	3.0	Yes	Returns 128.
<code>int getMaxStatementLength()</code>	3.0	Yes	Returns 0.
<code>int getMaxStatements()</code>	3.0	Yes	Returns 0.
<code>int getMaxTableNameLength()</code>	3.0	Yes	Returns 128.
<code>int getMaxTablesInSelect()</code>	3.0	Yes	Returns 0.
<code>int getMaxUserNameLength()</code>	3.0	Yes	Returns 0.
<code>String getNumericFunctions()</code>	3.0	Yes	Returns the Numeric Functions list from the specification related to the JDBC version of the driver.
<code>ResultSet getPrimaryKeys(String catalog, String schema, String table)</code>	3.0	Yes	
<code>ResultSet getProcedureColumns(String catalog, String schemaPattern, String procedureNamePattern, String columnNamePattern)</code>	3.0	Yes	
<code>ResultSet getProcedures(String</code>	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>catalog, String schemaPattern, String procedureNamePattern)</code>			
<code>String getProcedureTerm()</code>	3.0	Yes	Returns procedure.
<code>ResultSet getPseudoColumns(String catalog, String schemaPattern, String tableNamePattern, String columnNamePattern)</code>	4.1	Yes	
<code>int getResultSetHoldability()</code>	3.0	Yes	Returns <code>CLOSE_CURSORS_AT_COMMIT</code> .
<code>RowIdLifetime getRowIdLifetime()</code>	4.0	Yes	Returns <code>ROWID_UNSUPPORTED</code> .
<code>ResultSet getSchemas()</code>	3.0	Yes	
<code>ResultSet getSchemas(String catalog, String schemaPattern)</code>	4.0	Yes	
<code>String getSchemaTerm()</code>	3.0	Yes	Returns schema.
<code>String getSearchStringEscape()</code>	3.0	Yes	Returns a backslash (<code>\</code>).
<code>String getSQLKeywords()</code>	3.0	Yes	Returns an empty String.
<code>int getSQLStateType()</code>	3.0	Yes	Returns <code>sqlStateSQL99</code> .
<code>String getStringFunctions()</code>	3.0	Yes	Returns the String Functions list from the specification related to the JDBC version of the driver.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>ResultSet getSuperTables(String catalog, String schemaPattern, String tableNamePattern)</code>	3.0	Yes	
<code>ResultSet getSuperTypes(String catalog, String schemaPattern, String typeNamePattern)</code>	3.0	Yes	
<code>String getSystemFunctions()</code>	3.0	Yes	Returns DATABASE, IFNULL, USER.
<code>ResultSet getTablePrivileges(String catalog, String schemaPattern, String tableNamePattern)</code>	3.0	Yes	
<code>ResultSet getTables(String catalog, String schemaPattern, String tableNamePattern, String[] types)</code>	3.0	Yes	
<code>ResultSet getTableTypes()</code>	3.0	Yes	
<code>String getTimeDateFunctions()</code>	3.0	Yes	Returns the Time and Date Functions list from the specification related to the JDBC version of the driver.
<code>ResultSet getTypeInfo()</code>	3.0	Yes	
<code>ResultSet getUDTs(String catalog, String schemaPattern, String typeNamePattern, int[] types)</code>	3.0	Yes	
<code>String getURL()</code>	3.0	Yes	
<code>String getUsername()</code>	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>ResultSet getVersionColumns(String catalog, String schema, String table)</code>	3.0	Yes	
<code>boolean insertsAreDetected(int type)</code>	3.0	Yes	
<code>boolean isCatalogAtStart()</code>	3.0	Yes	
<code>boolean isReadOnly()</code>	3.0	Yes	Returns true.
<code>boolean locatorsUpdateCopy()</code>	3.0	Yes	Returns false.
<code>boolean nullPlusNonNullIsNull()</code>	3.0	Yes	Returns true.
<code>boolean nullsAreSortedAtEnd()</code>	3.0	Yes	Returns false.
<code>boolean nullsAreSortedAtStart()</code>	3.0	Yes	Returns false.
<code>boolean nullsAreSortedHigh()</code>	3.0	Yes	Returns false.
<code>boolean nullsAreSortedLow()</code>	3.0	Yes	Returns true.
<code>boolean othersDeletesAreVisible(int type)</code>	3.0	Yes	
<code>boolean othersInsertsAreVisible(int type)</code>	3.0	Yes	
<code>boolean othersUpdatesAreVisible(int type)</code>	3.0	Yes	
<code>boolean ownDeletesAreVisible(int type)</code>	3.0	Yes	
<code>boolean ownInsertsAreVisible(int type)</code>	3.0	Yes	
<code>boolean ownUpdatesAreVisible(int type)</code>	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>boolean storesLowerCaseIdentifiers()</code>	3.0	Yes	Returns false.
<code>boolean storesLowerCaseQuotedIdentifiers()</code>	3.0	Yes	Returns false.
<code>boolean storesMixedCaseIdentifiers()</code>	3.0	Yes	Returns true.
<code>boolean storesMixedCaseQuotedIdentifiers()</code>	3.0	Yes	Returns true.
<code>boolean storesUpperCaseIdentifiers()</code>	3.0	Yes	Returns false.
<code>boolean storesUpperCaseQuotedIdentifiers()</code>	3.0	Yes	Returns false.
<code>boolean supportsAlterTableWithAddColumn()</code>	3.0	Yes	Returns false.
<code>boolean supportsAlterTableWithDropColumn()</code>	3.0	Yes	Returns false.
<code>boolean supportsANSI92EntryLevelSQL()</code>	3.0	Yes	Returns true.
<code>boolean supportsANSI92FullSQL()</code>	3.0	Yes	Returns false.
<code>boolean supportsANSI92IntermediateSQL()</code>	3.0	Yes	Returns false.
<code>boolean supportsBatchUpdates()</code>	3.0	Yes	Returns false.
<code>boolean supportsCatalogsInDataManipulation()</code>	3.0	Yes	Returns true.
<code>boolean supportsCatalogsInIndexDefinitions()</code>	3.0	Yes	Returns true.
<code>boolean supportsCatalogsInPrivilegeDefinitions()</code>	3.0	Yes	Returns true.

Features

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>boolean supportsCatalogsInProcedureCalls()</code>	3.0	Yes	Returns true.
<code>boolean supportsCatalogsInTableDefinitions()</code>	3.0	Yes	Returns true.
<code>boolean supportsColumnAliasing()</code>	3.0	Yes	Returns true.
<code>boolean supportsConvert()</code>	3.0	Yes	Returns true.
<code>boolean supportsConvert(int fromType, int toType)</code>	3.0	Yes	
<code>boolean supportsCoreSQLGrammar()</code>	3.0	Yes	Returns true.
<code>boolean supportsCorrelatedSubqueries()</code>	3.0	Yes	Returns true.
<code>boolean supportsDataDefinitionAndDataManipulationTransactions()</code>	3.0	Yes	Returns false.
<code>boolean supportsDataManipulationTransactionsOnly()</code>	3.0	Yes	Returns false.
<code>boolean supportsDifferentTableCorrelationNames()</code>	3.0	Yes	Returns false.
<code>boolean supportsExpressionsInOrderBy()</code>	3.0	Yes	Returns true.
<code>boolean supportsExtendedSQLGrammar()</code>	3.0	Yes	Returns false.
<code>boolean supportsFullOuterJoins()</code>	3.0	Yes	Returns true.
<code>boolean supportsGetGeneratedKeys()</code>	3.0	Yes	Returns false.
<code>boolean supportsGroupBy()</code>	3.0	Yes	Returns true.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>boolean supportsGroupByBeyondSelect()</code>	3.0	Yes	Returns true.
<code>boolean supportsGroupByUnrelated()</code>	3.0	Yes	Returns false.
<code>boolean supportsIntegrityEnhancementFacility()</code>	3.0	Yes	Returns false.
<code>boolean supportsLikeEscapeClause()</code>	3.0	Yes	Returns true.
<code>boolean supportsLimitedOuterJoins()</code>	3.0	Yes	Returns false.
<code>boolean supportsMinimumSQLGrammar()</code>	3.0	Yes	Returns true.
<code>boolean supportsMixedCaseIdentifiers()</code>	3.0	Yes	Returns false.
<code>boolean supportsMixedCaseQuotedIdentifiers()</code>	3.0	Yes	Returns true.
<code>boolean supportsMultipleOpenResults()</code>	3.0	Yes	Returns false.
<code>boolean supportsMultipleResultSets()</code>	3.0	Yes	Returns false.
<code>boolean supportsMultipleTransactions()</code>	3.0	Yes	Returns true.
<code>boolean supportsNamedParameters()</code>	3.0	Yes	Returns false.
<code>boolean supportsNonNullableColumns()</code>	3.0	Yes	Returns false.
<code>boolean supportsOpenCursorsAcrossCommit()</code>	3.0	Yes	Returns false.
<code>boolean supportsOpenCursorsAcrossRollback()</code>	3.0	Yes	Returns false.
<code>boolean</code>	3.0	Yes	Returns true.

Features

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>supportsOpenStatementsAcrossCommit()</code>			
<code>boolean supportsOpenStatementsAcrossRollback()</code>	3.0	Yes	Returns true.
<code>boolean supportsOrderByUnrelated()</code>	3.0	Yes	Returns false.
<code>boolean supportsOuterJoins()</code>	3.0	Yes	Returns false.
<code>boolean supportsPositionedDelete()</code>	3.0	Yes	Returns false.
<code>boolean supportsPositionedUpdate()</code>	3.0	Yes	Returns false.
<code>boolean supportsResultSetConcurrency(int type, int concurrency)</code>	3.0	Yes	
<code>boolean supportsResultSetHoldability(int holdability)</code>	3.0	Yes	
<code>boolean supportsResultSetType(int type)</code>	3.0	Yes	
<code>boolean supportsSavepoints()</code>	3.0	Yes	Returns false.
<code>boolean supportsSchemasInDataManipulation()</code>	3.0	Yes	Returns true.
<code>boolean supportsSchemasInIndexDefinitions()</code>	3.0	Yes	Returns true.
<code>boolean supportsSchemasInPrivilegeDefinitions()</code>	3.0	Yes	Returns true.
<code>boolean supportsSchemasInProcedureCalls()</code>	3.0	Yes	Returns false.
<code>boolean supportsSchemasInTableDefinitions()</code>	3.0	Yes	Returns true.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>boolean supportsSelectForUpdate()</code>	3.0	Yes	Returns false.
<code>boolean supportsStatementPooling()</code>	3.0	Yes	Returns false.
<code>boolean supportsStoredFunctionsUsingCallSyntax()</code>	4.0	Yes	Returns false.
<code>boolean supportsStoredProcedures()</code>	3.0	Yes	Returns true.
<code>boolean supportsSubqueriesInComparisons()</code>	3.0	Yes	Returns true.
<code>boolean supportsSubqueriesInExists()</code>	3.0	Yes	Returns true.
<code>boolean supportsSubqueriesInIns()</code>	3.0	Yes	Returns true.
<code>boolean supportsSubqueriesInQuantifieds()</code>	3.0	Yes	Returns true.
<code>boolean supportsTableCorrelationNames()</code>	3.0	Yes	Returns true.
<code>boolean supportsTransactionIsolationLevel(int level)</code>	3.0	Yes	
<code>boolean supportsTransactions()</code>	3.0	Yes	Returns false.
<code>boolean supportsUnion()</code>	3.0	Yes	Returns true.
<code>boolean supportsUnionAll()</code>	3.0	Yes	Returns true.
<code>boolean updatesAreDetected(int type)</code>	3.0	Yes	Returns true.
<code>boolean usesLocalFilePerTable()</code>	3.0	Yes	Returns false.
<code>boolean usesLocalFiles()</code>	3.0	Yes	Returns false.

Features

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>boolean isWrapperFor(Class<?> iface)</code>	4.0	Yes	
<code><T> T unwrap(Class<T> iface)</code>	4.0	Yes	

DataSource

The following table lists the methods that belong to the `DataSource` interface, and describes whether each method is supported by the Cloudera JDBC Driver for Apache Hive and which version of the JDBC API is the earliest version that supports the method.

For detailed information about each method in the `DataSource` interface, see the Java API documentation: <http://docs.oracle.com/javase/1.5.0/docs/api/javax/sql/DataSource.html>.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>Connection getConnection()</code>	3.0	Yes	
<code>Connection getConnection(String username, String password)</code>	3.0	Yes	
<code>int getLoginTimeout()</code>	3.0	Yes	
<code>PrintWriter getLogWriter()</code>	3.0	Yes	
<code>Logger getParentLogger()</code>	4.1	No	The driver does not use <code>java.util.logging</code> .
<code>void setLoginTimeout(int seconds)</code>	3.0	Yes	
<code>void setLogWriter(PrintWriter out)</code>	3.0	Yes	
<code>boolean isWrapperFor</code>	4.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
(Class<?> iface)			
<T> T unwrap(Class<T> iface)	4.0	Yes	

Driver

The following table lists the methods that belong to the `Driver` interface, and describes whether each method is supported by the Cloudera JDBC Driver for Apache Hive and which version of the JDBC API is the earliest version that supports the method.

For detailed information about each method in the `Driver` interface, see the Java API documentation: <http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/Driver.html>.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
boolean acceptsURL (String url)	3.0	Yes	
Connection connect (String url, Properties info)	3.0	Yes	
int getMajorVersion()	3.0	Yes	
int getMinorVersion()	3.0	Yes	
Logger getParentLogger()	4.1	No	
DriverPropertyInfo[] getPropertyInfo (String url, Properties info)	3.0	Yes	
boolean jdbcCompliant()	3.0	Yes	

ParameterMetaData

The following table lists the methods that belong to the `ParameterMetaData` interface, and describes whether each method is supported by the Cloudera JDBC Driver for Apache Hive and which version of the JDBC API is the earliest version that supports the method.

For detailed information about each method in the `ParameterMetaData` interface, see the Java API documentation:

<http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/ParameterMetaData.html>.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>String getParameterClassName (int param)</code>	3.0	Yes	
<code>int getParameterCount()</code>	3.0	Yes	
<code>int getParameterMode(int param)</code>	3.0	Yes	
<code>int getParameterType(int param)</code>	3.0	Yes	
<code>String getParameterTypeName(int param)</code>	3.0	Yes	
<code>int getPrecision(int param)</code>	3.0	Yes	
<code>int getScale(int param)</code>	3.0	Yes	
<code>int isNullable(int param)</code>	3.0	Yes	
<code>boolean isSigned(int param)</code>	3.0	Yes	
<code>boolean isWrapperFor (Class<?> iface)</code>	4.0	Yes	
<code><T> T unwrap(Class<T> iface)</code>	4.0	Yes	

PooledConnection

The following table lists the methods that belong to the `PooledConnection` interface, and describes whether each method is supported by the Cloudera JDBC Driver for Apache Hive and which version of the JDBC API is the earliest version that supports the method.

For detailed information about each method in the `PooledConnection` interface, see the Java API documentation:

<http://docs.oracle.com/javase/1.5.0/docs/api/javax/sql/PooledConnection.html>.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>void addConnectionEventListener(ConnectionEventListener listener)</code>	3.0	Yes	
<code>void addStatementEventListener(StatementEventListener listener)</code>	4.0	Yes	
<code>void close()</code>	3.0	Yes	
<code>Connection getConnection()</code>	3.0	Yes	
<code>void removeConnectionEventListener(ConnectionEventListener listener)</code>	3.0	Yes	
<code>void removeStatementEventListener(StatementEventListener listener)</code>	4.0	Yes	Removes the specified <code>StatementEventListener</code> from the list of components that will be notified when the driver detects that a <code>PreparedStatement</code> has been closed or is invalid.

PreparedStatement

The `PreparedStatement` interface extends the `Statement` interface.

The following table lists the methods that belong to the `PreparedStatement` interface, and describes whether each method is supported by the Cloudera JDBC Driver for Apache Hive and which version of the JDBC API is the earliest version that supports the method.

For detailed information about each method in the `PooledConnection` interface, see the Java API documentation:

<http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/PreparedStatement.html>.

Features

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>void addBatch()</code>	3.0	Yes	
<code>void clearParameters()</code>	3.0	Yes	
<code>boolean execute()</code>	3.0	Yes	
<code>ResultSet executeQuery()</code>	3.0	Yes	
<code>int executeUpdate()</code>	3.0	Yes	
<code>ResultSetMetaData getMetaData()</code>	3.0	Yes	
<code>ParameterMetaData getParameterMetaData()</code>	3.0	Yes	
<code>void setArray(int parameterIndex, Array x)</code>	3.0	No	
<code>void setAsciiStream(int parameterIndex, InputStream x)</code>	4.0	Yes	
<code>void setAsciiStream(int parameterIndex, InputStream x, int length)</code>	3.0	Yes	
<code>void setAsciiStream(int parameterIndex, InputStream x, long length)</code>	4.0	Yes	
<code>void setBigDecimal(int parameterIndex, BigDecimal x)</code>	3.0	Yes	
<code>void setBinaryStream(int parameterIndex, InputStream x)</code>	4.0	Yes	
<code>void setBinaryStream(int parameterIndex,</code>	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>InputStream x, int length)</code>			
<code>void setBinaryStream(int parameterIndex, InputStream x, long length)</code>	4.0	Yes	
<code>void setBlob(int parameterIndex, Blob x)</code>	3.0	No	
<code>void setBlob(int parameterIndex, InputStream inputStream)</code>	4.0	No	
<code>void setBlob(int parameterIndex, InputStream inputStream, long length)</code>	4.0	No	
<code>void setBoolean(int parameterIndex, boolean x)</code>	3.0	Yes	
<code>void setByte(int parameterIndex, byte x)</code>	3.0	Yes	
<code>void setBytes(int parameterIndex, byte[] x)</code>	3.0	Yes	
<code>void setCharacterStream(int parameterIndex, Reader reader)</code>	4.0	Yes	
<code>void setCharacterStream(int parameterIndex, Reader reader, int length)</code>	3.0	Yes	
<code>void setCharacterStream(int parameterIndex, Reader reader, long length)</code>	4.0	Yes	

Features

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>void setClob(int parameterIndex, Clob x)</code>	3.0	No	
<code>void setClob(int parameterIndex, Reader reader)</code>	4.0	No	
<code>void setClob(int parameterIndex, Reader reader, long length)</code>	4.0	No	
<code>void setDate(int parameterIndex, Date x)</code>	3.0	Yes	
<code>void setDate(int parameterIndex, Date x, Calendar cal)</code>	3.0	Yes	
<code>void setDouble(int parameterIndex, double x)</code>	3.0	Yes	
<code>void setFloat(int parameterIndex, float x)</code>	3.0	Yes	
<code>void setInt(int parameterIndex, int x)</code>	3.0	Yes	
<code>void setLong(int parameterIndex, long x)</code>	3.0	Yes	
<code>void setNCharacterStream(int parameterIndex, Reader value)</code>	4.0	No	
<code>void setNCharacterStream(int parameterIndex, Reader value, long length)</code>	4.0	No	
<code>void setNClob(int parameterIndex, NClob value)</code>	4.0	No	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>void setNClob(int parameterIndex, Reader reader)</code>	4.0	No	
<code>void setNClob(int parameterIndex, Reader reader, long length)</code>	4.0	No	
<code>void setNString(int parameterIndex, String value)</code>	4.0	No	
<code>void setNull(int paramIndex, int sqlType, String typeName)</code>	3.0	Yes	
<code>void setObject(int parameterIndex, Object x)</code>	3.0	Yes	
<code>void setObject(int parameterIndex, Object x, int targetSqlType)</code>	3.0	Yes	
<code>void setObject(int parameterIndex, Object x, int targetSqlType, int scale)</code>	3.0	Yes	
<code>void setRef(int parameterIndex, Ref x)</code>	3.0	No	
<code>void setRowId(int parameterIndex, RowId x)</code>	4.0	No	
<code>void setShort(int parameterIndex, short x)</code>	3.0	No	
<code>void setSQLXML(int parameterIndex, SQLXML xmlObject)</code>	4.0	Yes	
<code>void setString(int</code>	3.0	Yes	

Features

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>parameterIndex, String x)</code>			
<code>void setTime(int parameterIndex, Time x)</code>	3.0	Yes	
<code>void setTime(int parameterIndex, Time x, Calendar cal)</code>	3.0	Yes	
<code>void setTimestamp(int parameterIndex, Timestamp x)</code>	3.0	Yes	
<code>void setTimestamp(int parameterIndex, Timestamp x, Calendar cal)</code>	3.0	Yes	
<code>void setUnicodeStream(int parameterIndex, InputStream x, int length)</code>	3.0	Yes	Deprecated.
<code>void setURL(int parameterIndex, URL x)</code>	3.0	No	
<code>boolean isWrapperFor(Class<?> iface)</code>	4.0	Yes	
<code><T> T unwrap(Class<T> iface)</code>	4.0	Yes	

ResultSet

The following table lists the methods that belong to the `ResultSet` interface, and describes whether each method is supported by the Cloudera JDBC Driver for Apache Hive and which version of the JDBC API is the earliest version that supports the method.

For detailed information about each method in the `ResultSet` interface, see the Java API documentation: <http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/ResultSet.html>.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>boolean absolute(int row)</code>	3.0	No	
<code>void afterLast()</code>	3.0	No	
<code>void beforeFirst()</code>	3.0	No	
<code>void cancelRowUpdates()</code>	3.0	No	Not valid because the driver is read-only.
<code>void clearWarnings()</code>	3.0	Yes	
<code>void close()</code>	3.0	Yes	
<code>void deleteRow()</code>	3.0	No	Not valid because the driver is read-only.
<code>int findColumn(String columnName)</code>	3.0	Yes	
<code>boolean first()</code>	3.0	No	
<code>Array getArray(int i)</code>	3.0	No	
<code>Array getArray(String colName)</code>	3.0	No	
<code>InputStream getAsciiStream(int columnIndex)</code>	3.0	Yes	
<code>InputStream getAsciiStream(String columnName)</code>	3.0	Yes	
<code>BigDecimal getBigDecimal(int columnIndex)</code>	3.0	Yes	
<code>BigDecimal getBigDecimal(int columnIndex, int scale)</code>	3.0	Yes	Deprecated.
<code>BigDecimal getBigDecimal</code>	3.0	Yes	

Features

Method	Supported Since JDBC Version	Supported by the Driver	Notes
(String columnName)			
BigDecimal getBigDecimal (String columnName, int scale)	3.0	Yes	Deprecated.
InputStream getBinaryStream(int columnIndex)	3.0	Yes	
InputStream getBinaryStream(String columnName)	3.0	Yes	
Blob getBlob(int i)	3.0	No	
Blob getBlob(String colName)	3.0	No	
boolean getBoolean(int columnIndex)	3.0	Yes	
boolean getBoolean(String columnName)	3.0	Yes	
getBytes(int columnIndex)	3.0	Yes	
byte getByte(String columnName)	3.0	Yes	
byte[] getBytes(int columnIndex)	3.0	Yes	
byte[] getBytes(String columnName)	3.0	Yes	
Reader getCharacterStream (int columnIndex)	3.0	Yes	
Reader getCharacterStream (String columnName)	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
Clob getClob(int i)	3.0	No	
Clob getClob(String colName)	3.0	No	
int getConcurrency()	3.0	Yes	
String getCursorName()	3.0	Yes	
Date getDate(int columnIndex)	3.0	Yes	
Date getDate(int columnIndex, Calendar cal)	3.0	Yes	
Date getDate(String columnName)	3.0	Yes	
Date getDate(String columnName, Calendar cal)	3.0	Yes	
double getDouble(int columnIndex)	3.0	Yes	
double getDouble(String columnName)	3.0	Yes	
int getFetchDirection()	3.0	Yes	
int getFetchSize()	3.0	Yes	
float getFloat(int columnIndex)	3.0	Yes	
float getFloat(String columnName)	3.0	Yes	
int getHoldability()	4.0	Yes	
int getInt(int columnIndex)	3.0	Yes	

Features

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>int getInt(String columnName)</code>	3.0	Yes	
<code>long getLong(int columnIndex)</code>	3.0	Yes	
<code>long getLong(String columnName)</code>	3.0	Yes	
<code>ResultSetMetaData getMetaData()</code>	3.0	Yes	
<code>Reader getNCharacterStream(int columnIndex)</code>	4.0	No	
<code>Reader getNCharacterStream(String columnName)</code>	4.0	No	
<code>NClob getNClob(int columnIndex)</code>	4.0	No	
<code>NClob getNClob(String columnName)</code>	4.0	No	
<code>String getNString(int columnIndex)</code>	4.0	No	
<code>String getNString(String columnName)</code>	4.0	No	
<code>Object getObject(int columnIndex)</code>	3.0	Yes	
<code><T> T getObject(int columnIndex, Class<T> type)</code>	4.1	No	
<code>Object getObject(int i, Map<String, Class<?>> map)</code>	3.0	No	
<code>Object getObject(String columnName)</code>	3.0	No	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
columnName)			
<T> T getObject(String columnName, Class<T> type)	4.1	No	
Object getObject(String colName, Map<String,Class<?>> map)	3.0	Yes	
Ref getRef(int i)	3.0	No	
Ref getRef(String colName)	3.0	No	
int getRow()	3.0	Yes	
RowId getRowId(int columnIndex)	4.0	No	
RowId getRowId(String columnLabel)	4.0	No	
short getShort(int columnIndex)	3.0	Yes	
short getShort(String columnName)	3.0	Yes	
SQLXML getSQLXML(int columnIndex)	4.0	No	
SQLXML getSQLXML(String columnLabel)	4.0	No	
Statement getStatement()	3.0	Yes	
String getString(int columnIndex)	3.0	Yes	
String getString(String columnName)	3.0	Yes	

Features

Method	Supported Since JDBC Version	Supported by the Driver	Notes
Time getTime(int columnIndex)	3.0	Yes	
Time getTime(int columnIndex, Calendar cal)	3.0	Yes	
Time getTime(String columnName)	3.0	Yes	
Time getTime(String columnName, Calendar cal)	3.0	Yes	
Timestamp getTimestamp(int columnIndex)	3.0	Yes	
Timestamp getTimestamp(int columnIndex, Calendar cal)	3.0	Yes	
Timestamp getTimestamp(String columnName)	3.0	Yes	
Timestamp getTimestamp(String columnName, Calendar cal)	3.0	Yes	
int getType()	3.0	Yes	
InputStream getUnicodeStream(int columnIndex)	3.0	Yes	Deprecated.
InputStream getUnicodeStream(String columnName)	3.0	Yes	Deprecated.
URL getURL(int columnIndex)	3.0	No	
URL getURL(String columnName)	3.0	No	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
SQLWarning getWarnings()	3.0	Yes	
void insertRow()	3.0	No	Not valid because the driver is read-only.
boolean isAfterLast()	3.0	Yes	
boolean isBeforeFirst()	3.0	Yes	
boolean isClosed()	4.0	Yes	
boolean isFirst()	3.0	Yes	
boolean isLast()	3.0	No	
boolean last()	3.0	No	
void moveToCurrentRow()	3.0	No	Not valid because the driver is read-only.
void moveToInsertRow()	3.0	No	Not valid because the driver is read-only.
boolean next()	3.0	Yes	
boolean previous()	3.0	No	
void refreshRow()	3.0	No	
boolean relative(int rows)	3.0	No	
boolean rowDeleted()	3.0	Yes	Hard-coded to false.
boolean rowInserted()	3.0	Yes	Hard-coded to false.
boolean rowUpdated()	3.0	Yes	Hard-coded to false.
void setFetchDirection(int direction)	3.0	No	Not valid because the driver is forward-only.

Features

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>void setFetchSize(int rows)</code>	3.0	Yes	
<code>void updateArray(int columnIndex, Array x)</code>	3.0	No	
<code>void updateArray(String columnName, Array x)</code>	3.0	No	
<code>void updateAsciiStream(int columnIndex, InputStream x)</code>	4.0	No	Not valid because the driver is read-only.
<code>void updateAsciiStream(int columnIndex, InputStream x, int length)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateAsciiStream(int columnIndex, InputStream x, long length)</code>	4.0	No	Not valid because the driver is read-only.
<code>void updateAsciiStream(String columnName, InputStream x)</code>	4.0	No	Not valid because the driver is read-only.
<code>void updateAsciiStream(String columnName, InputStream x, int length)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateAsciiStream(String columnName, InputStream x, long length)</code>	4.0	No	Not valid because the driver is read-only.
<code>void updateBigDecimal(int columnIndex, BigDecimal x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateBigDecimal(String columnName, BigDecimal x)</code>	3.0	No	Not valid because the driver is read-only.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>void updateBinaryStream(int columnIndex, InputStream x)</code>	4.0	No	Not valid because the driver is read-only.
<code>void updateBinaryStream(int columnIndex, InputStream x, int length)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateBinaryStream(int columnIndex, InputStream x, long length)</code>	4.0	No	Not valid because the driver is read-only.
<code>void updateBinaryStream(String columnName, InputStream x)</code>	4.0	No	Not valid because the driver is read-only.
<code>void updateBinaryStream(String columnName, InputStream x, int length)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateBinaryStream(String columnName, InputStream x, long length)</code>	4.0	No	Not valid because the driver is read-only.
<code>void updateBlob(int columnIndex, InputStream inputStream)</code>	4.0	No	
<code>void updateBlob(int columnIndex, Blob x)</code>	3.0	No	
<code>void updateBlob(int columnIndex, InputStream inputStream, long length)</code>	4.0	No	
<code>void updateBlob(String columnName, InputStream inputStream)</code>	4.0	No	
<code>void updateBlob(String</code>	3.0	No	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>columnName, Blob x)</code>			
<code>void updateBlob(String columnLabel, InputStream inputStream, long length)</code>	4.0	No	
<code>void updateBoolean(int columnIndex, boolean x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateBoolean(String columnName, boolean x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateByte(int columnIndex, byte x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateByte(String columnName, byte x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateBytes(int columnIndex, byte[] x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateBytes(String columnName, byte[] x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateCharacterStream(int columnIndex, Reader x, int length)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateCharacterStream(String columnName, Reader reader, int length)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateBlob(int columnIndex, InputStream inputStream)</code>	4.0	No	
<code>void updateClob(int columnIndex, Clob x)</code>	3.0	No	
<code>void updateBlob(int</code>	4.0	No	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>columnIndex, InputStream inputStream, long length)</code>			
<code>void updateBlob(String columnName, InputStream inputStream)</code>	4.0	No	
<code>void updateClob(String columnName, Clob x)</code>	3.0	No	
<code>void updateBlob(String columnName, InputStream inputStream, long length)</code>	4.0	No	
<code>void updateDate(int columnIndex, Date x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateDate(String columnName, Date x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateDouble(int columnIndex, double x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateDouble(String columnName, double x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateFloat(int columnIndex, float x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateFloat(String columnName, float x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateInt(int columnIndex, int x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateInt(String columnName, int x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateLong(int columnIndex, long x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateLong(String</code>	3.0	No	Not valid because the driver

Features

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>columnName, long x)</code>			is read-only.
<code>void updateNCharacterStream (int columnIndex, Reader x)</code>	4.0	No	
<code>void updateNCharacterStream (int columnIndex, Reader x, long length)</code>	4.0	No	
<code>void updateNCharacterStream (String columnName, Reader reader)</code>	4.0	No	
<code>void updateNCharacterStream (String columnName, Reader reader, long length)</code>	4.0	No	
<code>void updateNClob(int columnIndex, NClob nClob)</code>	4.0	No	
<code>void updateNClob(int columnIndex, Reader reader)</code>	4.0	No	
<code>void updateNClob(int columnIndex, Reader reader, long length)</code>	4.0	No	
<code>void updateNClob(String columnName, NClob nClob)</code>	4.0	No	
<code>void updateNClob(String columnName, Reader reader)</code>	4.0	No	
<code>void updateNClob(String columnName, Reader reader, long length)</code>	4.0	No	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>void updateNString(int columnIndex, String nString)</code>	4.0	No	
<code>void updateNString(String columnName, String nString)</code>	4.0	No	
<code>void updateNull(int columnIndex)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateNull(String columnName)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateObject(int columnIndex, Object x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateObject(int columnIndex, Object x, int scale)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateObject(String columnName, Object x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateObject(String columnName, Object x, int scale)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateRef(int columnIndex, Ref x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateRef(String columnName, Ref x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateRow()</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateRowId(int columnIndex, RowId x)</code>	4.0	No	
<code>void updateRowId(String columnName, RowId x)</code>	4.0	No	

Features

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>void updateShort(int columnIndex, short x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateShort(String columnName, short x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateSQLXML(int columnIndex, SQLXML xmlObject)</code>	4.0	No	
<code>void updateSQLXML(String columnName, SQLXML xmlObject)</code>	4.0	No	
<code>void updateString(int columnIndex, String x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateString(String columnName, String x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateTime(int columnIndex, Time x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateTime(String columnName, Time x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateTimeStamp(int columnIndex, Timestamp x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateTimeStamp(String columnName, Timestamp x)</code>	3.0	No	Not valid because the driver is read-only.
<code>boolean wasNull()</code>	3.0	Yes	
<code>boolean isWrapperFor(Class<?> iface)</code>	4.0	Yes	
<code><T> T unwrap(Class<T> iface)</code>	4.0	Yes	

ResultSetMetaData

The following table lists the methods that belong to the `ResultSetMetaData` interface, and describes whether each method is supported by the Cloudera JDBC Driver for Apache Hive and which version of the JDBC API is the earliest version that supports the method.

For detailed information about each method in the `ResultSetMetaData` interface, see the Java API documentation:

<http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/ResultSetMetaData.html>.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>String getCatalogName (int column)</code>	3.0	Yes	
<code>String getColumnClassName (int column)</code>	3.0	Yes	
<code>int getColumnCount ()</code>	3.0	Yes	
<code>int getColumnDisplaySize (int column)</code>	3.0	Yes	
<code>String getColumnLabel (int column)</code>	3.0	Yes	
<code>String getColumnName (int column)</code>	3.0	Yes	
<code>int getColumnType (int column)</code>	3.0	Yes	
<code>String getColumnTypeName (int column)</code>	3.0	Yes	
<code>int getPrecision (int column)</code>	3.0	Yes	
<code>int getScale (int column)</code>	3.0	Yes	
<code>String getSchemaName (int column)</code>	3.0	Yes	
<code>String getTableName (int column)</code>	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>boolean isAutoIncrement(int column)</code>	3.0	Yes	
<code>boolean isCaseSensitive(int column)</code>	3.0	Yes	
<code>boolean isCurrency(int column)</code>	3.0	Yes	
<code>boolean isDefinitelyWritable(int column)</code>	3.0	Yes	
<code>int isNullable(int column)</code>	3.0	Yes	
<code>boolean isReadOnly(int column)</code>	3.0	Yes	
<code>boolean isSearchable(int column)</code>	3.0	Yes	
<code>boolean isSigned(int column)</code>	3.0	Yes	
<code>boolean isWritable(int column)</code>	3.0	Yes	
<code>boolean isWrapperFor(Class<?> iface)</code>	4.0	Yes	
<code><T> T unwrap(Class<T> iface)</code>	4.0	Yes	

Statement

The following table lists the methods that belong to the `Statement` interface, and describes whether each method is supported by the Cloudera JDBC Driver for Apache Hive and which version of the JDBC API is the earliest version that supports the method.

For detailed information about each method in the `Statement` interface, see the Java API documentation: <http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/Statement.html>.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>void addBatch(String sql)</code>	3.0	Yes	
<code>void cancel()</code>	3.0	Yes	
<code>void clearBatch()</code>	3.0	Yes	
<code>void clearWarnings()</code>	3.0	Yes	
<code>void close()</code>	3.0	Yes	
<code>void closeOnCompletion()</code>	4.1	Yes	
<code>boolean execute(String sql)</code>	3.0	Yes	
<code>boolean execute(String sql, int autoGeneratedKeys)</code>	3.0	No	
<code>boolean execute(String sql, int[] columnIndexes)</code>	3.0	No	
<code>boolean execute(String sql, String[] columnNames)</code>	3.0	No	
<code>int[] executeBatch()</code>	3.0	No	
<code>ResultSet executeQuery(String sql)</code>	3.0	Yes	
<code>int executeUpdate(String sql)</code>	3.0	Yes	
<code>int executeUpdate(String sql, int autoGeneratedKeys)</code>	3.0	No	
<code>int executeUpdate(String sql, int[] columnIndexes)</code>	3.0	No	
<code>int executeUpdate(String sql, String[] columnNames)</code>	3.0	No	

Features

Method	Supported Since JDBC Version	Supported by the Driver	Notes
columnNames)			
Connection getConnection()	3.0	Yes	
int getFetchDirection()	3.0	Yes	
int getFetchSize()	3.0	Yes	
ResultSet getGeneratedKeys()	3.0	Yes	
int getMaxFieldSize()	3.0	Yes	
int getMaxRows()	3.0	Yes	
boolean getMoreResults()	3.0	Yes	
boolean getMoreResults(int current)	3.0	No	
int getQueryTimeout()	3.0	Yes	
ResultSet getResultSet()	3.0	Yes	
int getResultSetConcurrency()	3.0	Yes	Hard-coded to CONCUR_READ_ONLY.
int getResultSetHoldability()	3.0	Yes	Hard-coded to CLOSE_CURSORS_AT_COMMIT.
int getResultSetType()	3.0	Yes	Hard-coded to TYPE_FORWARD_ONLY.
int getUpdateCount()	3.0	Yes	
SQLWarning getWarnings()	3.0	Yes	
boolean isClosed()	4.0	Yes	
boolean isCloseOnCompletion()	4.1	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
<code>boolean isPoolable()</code>	4.0	Yes	
<code>void setCursorName(String name)</code>	3.0	No	
<code>void setEscapeProcessing(boolean enable)</code>	3.0	Yes	
<code>void setFetchDirection(int direction)</code>	3.0	No	
<code>void setFetchSize(int rows)</code>	3.0	Yes	
<code>void setMaxFieldSize(int max)</code>	3.0	Yes	
<code>void setMaxRows(int max)</code>	3.0	Yes	
<code>void setPoolable(boolean poolable)</code>	4.0	Yes	
<code>void setQueryTimeout(int seconds)</code>	3.0	Yes	
<code>boolean isWrapperFor(Class<?> iface)</code>	4.0	Yes	
<code><T> T unwrap(Class<T> iface)</code>	4.0	Yes	

Contact Us

If you are having difficulties using the driver, our [Community Forum](#) may have your solution. In addition to providing user to user support, our forums are a great place to share your questions, comments, and feature requests with us.

If you are a Subscription customer you may also use the [Cloudera Support Portal](#) to search the Knowledge Base or file a Case.

Important:

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

Driver Configuration Options

Driver Configuration Options lists and describes the properties that you can use to configure the behavior of the Cloudera JDBC Driver for Apache Hive.

You can set configuration properties using the connection URL. For more information, see "Building the Connection URL" on page 10.

Note:

Property names and values are case-sensitive.

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.

Important:

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

Note:

This property is applicable only when SSL connections are enabled.

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.

AuthMech

Default Value	Data Type	Required
Depends on the <code>transportMode</code> setting. For more information, see "transportMode" on page 93.	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.

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 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.

DefaultStringLength

Default Value	Data Type	Required
255	Integer	No

Description

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

Note:

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=cloudera;
```

Note:

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 <code>AuthMech=1</code> .

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.

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 `HiveJDBC_driver.log` file that logs driver activity that is not specific to a connection.

Driver Configuration Options

- 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 <code>AuthMech=3</code> .

Description

The password corresponding to the user name that you provided using the property "UID" on page 93.

Important:

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.

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	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 92.

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 92.

SSLTrustStore

Default Value	Data Type	Required
<code>jssecacerts</code> , if it exists. If <code>jssecacerts</code> does not exist, then <code>cacerts</code> is used. The default location of <code>cacerts</code> is <code>jre\lib\security\</code> <code>None</code> .	String	No

Description

The full path and file name of the Java TrustStore containing the SSL certificates to use during authentication.

See also the property "SSLTrustStorePwd" on page 92.

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 92.

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.

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 88.

UID

Default Value	Data Type	Required
anonymous	String	Yes, if <code>AuthMech=3</code> .

Description

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

Important:

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.

Note:

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.