Cloudera ODBC Driver for Apache Hive
Version 2.5.12
## Table of Contents

INTRODUCTION.................................................................................................................................................. 1

WINDOWS DRIVER............................................................................................................................................. 1

  SYSTEM REQUIREMENTS .............................................................................................................................. 1
  INSTALLING THE DRIVER .............................................................................................................................. 2
  CONFIGURING ODBC CONNECTIONS ........................................................................................................... 2
  CONFIGURING AUTHENTICATION ................................................................................................................... 8
  Using No Authentication ....................................................................................................................................... 8
  Using Kerberos .................................................................................................................................................. 8
  Using User Name ............................................................................................................................................ 9
  Using User Name and Password ....................................................................................................................... 9
  Using User Name and Password (SSL) ........................................................................................................... 9
  Using Windows Azure HDInsight Emulator ................................................................................................... 10
  Using Windows Azure HDInsight Service ..................................................................................................... 10
  Using HTTP ................................................................................................................................................ 10
  Using HTTPS .................................................................................................................................................. 10
  Using Kerberos over HTTP ............................................................................................................................ 11
  Using Kerberos over HTTPS .......................................................................................................................... 12
  CONFIGURING DSN-LESS AUTHENTICATION ............................................................................................... 13

LINUX DRIVER ................................................................................................................................................... 14

  SYSTEM REQUIREMENTS .............................................................................................................................. 14
  INSTALLATION ............................................................................................................................................... 14
  Setting the LD_LIBRARY_PATH Environment Variable ................................................................................ 15

MAC OS X DRIVER ............................................................................................................................................. 16

  SYSTEM REQUIREMENTS .............................................................................................................................. 16
  INSTALLATION ............................................................................................................................................... 16
  Setting the DYLD_LIBRARY_PATH Environment Variable .......................................................................... 16

AIX DRIVER ....................................................................................................................................................... 17

  SYSTEM REQUIREMENTS .............................................................................................................................. 17
  INSTALLATION ............................................................................................................................................... 17
  SETTING THE LD_LIBRARY_PATH ENVIRONMENT VARIABLE ................................................................. 18
Introduction

Welcome to the Cloudera ODBC Driver for Hive. ODBC is one the most established and widely supported APIs for connecting to and working with databases. At the heart of the technology is the ODBC driver, which connects an application to the database.

Cloudera ODBC Driver for 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. Hive Query Language is a subset of SQL-92. If an application is Hive-aware, then the driver is configurable to pass the query through. 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, refer to the section “Features” on page 26.

Cloudera ODBC Driver for Hive is available for Microsoft Windows, Linux, and Mac OS X. It complies with the ODBC 3.52 data standard and adds important functionality such as Unicode and 32- and 64-bit support for high-performance computing environments on all platforms. Any version of the ODBC driver will connect to a Hive server irrespective of the server’s host OS.

This guide is suitable for users who are looking to access data residing within Hive from their desktop environment. Application developers may also find the information helpful. Refer to your application for details on connecting via ODBC.

Windows Driver

System Requirements

You install Cloudera ODBC Driver for Hive on client computers accessing data in a Hadoop cluster with the Hive service installed and running. Each computer where you install the driver must meet the following minimum system requirements:

- One of the following operating systems (32- and 64-bit editions are supported):
  - Windows® XP with SP3
  - Windows® Vista
  - Windows® 7 Professional
  - Windows® Server 2008 R2
- 25 MB of available disk space

The driver is suitable for use with all versions of Apache Hive.

Important:

To install the driver, you need Administrator privileges on the computer.
Windows Driver

Installing the Driver

On 64-bit Windows operating systems, you can execute 32- and 64-bit applications transparently. You must use the version of the driver matching the bitness of the client application accessing data in Hadoop / Hive:

- ClouderaHiveODBC32.msi for 32-bit applications
- ClouderaHiveODBC64.msi for 64-bit applications

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

Note:

To install Cloudera ODBC Driver for Hive:

1. Depending on the bitness of your client application, double-click to run ClouderaHiveODBC32.msi or ClouderaHiveODBC64.msi.
2. Click Next.
3. Select the check box to accept the terms of the License Agreement if you agree, and then click Next.
4. To change the installation location, click the Change button, then browse to the desired folder, and then click OK. To accept the installation location, click Next.
5. Click Install.
6. When the installation completes, click Finish.

Configuring ODBC Connections

To create a Data Source Name (DSN):

1. Click the Start button , then click All Programs, then click the Cloudera ODBC Driver for Apache Hive 2.5 (64-bit) or the Cloudera ODBC Driver for Apache Hive 2.5 (32-bit) program group.
   
   If you installed both versions of the driver, you will see two program groups. Because DSNs are bit-specific, select the version that matches the bitness of your application. For example, a DSN that is defined for the 32-bit driver will only be accessible from 32-bit applications.

2. Click 64-bit ODBC Administrator or 32-bit ODBC Administrator. The ODBC Data Source Administrator window opens.
3. Click the Drivers tab and verify that the Cloudera Hive ODBC Driver appears in the list of ODBC drivers that are installed on your system.
4. Click the System DSN tab to create a system DSN or click the User DSN tab to create a user DSN.
Note:
A system DSN is available for all users who log in to a workstation. A user DSN is available only to the user who creates the DSN.

5. Click Add.
The Create New Data Source window opens.

6. Select Cloudera ODBC Driver for Apache Hive and then click Finish.
The Cloudera Hive ODBC Driver DSN Setup window opens.

7. Use the options in the Cloudera Hive ODBC Driver DSN Setup window to configure your DSN:
   a. In the Data Source Name field, type a name for your DSN.
   b. Optionally, in the Description field, type a description.
   c. In the Host field, type the IP address or hostname of the Hive server.
   d. In the Port field, type the listening port for the service.
   e. In the Database field, type the name of the database schema to use when a schema is not explicitly specified in a query. Queries on other schemas can still be issued by explicitly specifying the schema in the query. To determine the appropriate database schema to use, type the show databases command at the Hive command prompt to inspect your databases.
   f. Click the drop-down arrow next to the Hive Server Type field and select either Hive Server 1 or Hive Server 2.
   g. Optionally, if the operations against Hive are to be done on behalf of a user that is different than the authenticated user for the connection, enter the user name of the user to be delegated in the Delegation UID text box.

Note:
This setting is only applicable when connecting to a Hive Server 2 that supports this feature. Otherwise this setting will not take any effect.

8. Optionally, if you selected Hive Server 2 as the Hive server type, you can configure authentication. For detailed instructions, refer to the section “Configuring Authentication” on page 8.

9. Optionally, in the Cloudera Hive ODBC Driver DSN Setup window, click Advanced Options to configure advanced options for your DSN:
   a. Select the Use Native Query check box to disable the SQL Connector feature.
Note:
The SQL Connector feature allows the driver to apply transformations to the queries emitted by an application to convert them into an equivalent form in HiveQL. If the application is Hive aware and already emits HiveQL, then turning off the SQL Connector feature avoids the extra overhead of query transformation.

b. Select the **Fast SQLPrepare** check box to defer query execution to SQLExecute.

Note:
When using Native Query mode, the driver will execute the HiveQL query to retrieve the result set metadata for SQLPrepare. As a result, SQLPrepare might be slow. If the result set metadata is not required after calling SQLPrepare, then enable this option.

c. Select the **Driver Config Take Precedence** check box to allow driver-wide configurations to take precedence over connection string and DSN settings.

d. Select the **Use Async Exec** check box to use the asynchronous version of the Hive client API call for executing a query.

Note:
This setting only takes effect when connecting to a Hive cluster running Hive 0.12.0 or higher.

e. Select the **Get Tables With Query** check box to retrieve the names of tables in a particular database using the GET TABLES query when retrieving metadata from Hive Server 2.
Note:
This setting is only applicable when connecting to Hive Server 2.

f. Select the **Unicode SQL character types** check box to enable the driver to return SQL_WVARCHAR instead of SQL_VARCHAR for STRING and VARCHAR columns, and SQL_WCHAR instead of SQL_CHAR for CHAR columns.

g. Select the **Show HIVE_SYSTEM Table** check box to enable the driver to return the HIVE_SYSTEM table for catalog function calls such as SQLTables and SQLColumns.

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

Note:
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.

i. In the **Default String Column Length** field, type the default string column length to use.

Note:
Hive does not provide the length for String columns in its column metadata. This option allows you to tune the length of String columns.

j. In the **Binary column length** field, type the maximum data length for binary columns.

Note:
Hive does not provide the maximum data length for binary columns in the columns metadata. The option allows you to tune the maximum data length for binary columns.

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

Note:
This setting only takes effect when connecting to a server running Hive earlier than 0.12.0. For Hive 0.12.0 and later, the driver uses the DECIMAL(p,s) semantic and the values for precision and scale are retrieved from the server.
1. In the **Async Exec Poll Interval** field, type the time in milliseconds between each poll of the query execution status.

**Note:**
Asynchronous query execution is only available starting with Hive 0.12.0.

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

**Note:**
This setting is only applicable to the **User Name and Password (SSL) authentication** mechanism and will be ignored by other authentication mechanisms.

n. In the **Trusted Certificate** field, enter the path to the file containing the list of trusted CA certificates in the PEM format.

**Note:**
This is only applicable to the **User Name and Password (SSL)** and **HTTPS** authentication mechanisms, and will be ignored by other authentication mechanisms.

**Note:**
SSL certificates in the trusted certificates file have to be in the **PEM** format.

**Note:**
If this setting is not set the driver will default to using the trusted CA certificates PEM file installed by the driver.

10. Optionally, in the Advanced Options window, click **Server Side Properties** to configure server-side properties for your DSN:

   a. To create a server-side property, click the **Add** button, then type appropriate values in the Key and Value fields, and then click **OK**

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

**OR**

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

---

**Note:**

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

---

b. If you selected Hive Server 2 as the Hive server type, then select or clear the **Apply Server Side Properties with Queries** check box as needed.

---

**Note:**

If you selected Hive Server 2, then the Apply Server Side Properties with Queries check box is selected by default. Selecting the check box configures the driver to apply each server-side property you set by executing a query when opening a session to the Hive server. Clearing the check box configures the driver to use a more efficient method to apply server-side properties that does not involve additional network round tripping. Some Hive Server 2 builds are not compatible with the more efficient method. If the server-side properties you set do not take effect when the check box is clear, then select the check box. If you selected Hive Server 1 as the Hive server type, then the **Apply Server Side Properties with Queries** check box is selected and unavailable.

---

c. Select the **Convert SSP Key Name to Lower Case** check box to force the driver to convert the server-side property key name to all lower case characters.

d. Click **OK**

11. Optionally, in the Advanced Options window, click **Temporary Table Configuration** to configure the Temporary Table feature for your DSN:

   a. Select the **Enable Temporary Table** check box to enable the temporary table feature.

   b. In the **Web HDFS Host** text box, enter the hostname or IP address of the machine hosting both the NameNode of your Hadoop cluster and the WebHDFS service. If this field is left blank the hostname of the Hive Server will be used.

   c. In the **Web HDFS Port** text box, enter the WebHDFS port for the NameNode.

   d. In the **HDFS User** text box, enter the name of the HDFS user that the driver will use to create the necessary files for supporting the Temporary Table feature.

   e. In the **Data file HDFS dir** text box, enter the HDFS directory that the driver will use to store the necessary files for supporting the Temporary Table feature.
f. In the Temp Table TTL text box, enter the number of minutes a temporary table is guaranteed to exist in Hive after it is created.

g. Click OK.

12. In the Advanced Options Window, click OK.

13. In the Cloudera Hive ODBC Driver DSN Setup window, click Test to test the connection, and then click OK.

For details on configuration options available to control the behavior of DSNs using Cloudera ODBC Driver for Hive, see “Appendix C: Driver Configuration Options” on page 38.

Configuring Authentication

For details on selecting the appropriate authentication for a DSN using Cloudera ODBC Driver for Hive, see “Appendix A: Authentication Options” on page 30. The authentication methods available are as follows:

- No Authentication
- Kerberos
- User Name
- User Name and Password
- HTTP
- HTTPS
- Kerberos over HTTP
- Kerberos over HTTPS

Using No Authentication

No additional details are required when using No Authentication.

Using Kerberos

To use Kerberos authentication, Kerberos must be configured prior to use. See “Appendix B: Configuring Kerberos Authentication for Windows” on page 34 for details.

After Kerberos has been installed and configured, then set the following options in the Authentication group in the Cloudera Hive ODBC Driver dialog box:

1. In the Mechanism field, select Kerberos.

2. If there is no default realm configured for your Kerberos setup, then type the value for the Kerberos realm of the Hive Server 2 host. Otherwise leave it blank. The Realm is only needed if
your Kerberos setup does not define a default realm or if the realm of your Hive Server 2 is not the default.

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

4. In the **Service Name** field, type the value for the service name of the Hive Server 2. For example, if the principle for the Hive Server 2 is "hive/fully.qualified.domain.name@YOUR-REALM.COM", then the value in the service name field should be **hive**. If you are unsure of the correct service name to use for your particular Hadoop deployment, see your Hadoop administrator.

**Using User Name**

For **User Name** authentication, select **User Name** in the **Mechanism** field in the **Cloudera Hive ODBC Driver** dialog box, and then type a user name in the **User Name** field.

**Using User Name and Password**

To configure your DSN for **User Name and Password** authentication:

1. In the Cloudera Hive ODBC Driver DSN Setup dialog, click the drop-down arrow next to the **Mechanism** field, and then select **User Name and Password**.

2. In the **User Name** field, type an appropriate credential.

3. In the **Password** field, type the password corresponding to the user name you typed in step 2.

**Using User Name and Password (SSL)**

To configure **User Name and Password (SSL)** authentication:

1. Click the drop-down arrow next to the **Mechanism** field, and then select **User Name and Password (SSL)**.

2. In the **User Name** field, type an appropriate credential.

3. In the **Password** field, type the password corresponding to the user name you typed in step 2.

---

**Note:**

The driver always accepts the use of self-signed SSL certificate for this authentication mechanism.

---

**Note:**

Optionally you can configure the **Allow Common Name Host Name Mismatch** setting to control whether the driver allows the common name of a CA issued certificate to not match the host name of the Hive server. For self-signed certificates the driver always allows the common name of the certificate to not match the host name.
Windows Driver

Note:
Optionally, you can configure the Trusted Certificates setting to specify the file listing the SSL certificate authorities (CAs) you would like the driver to trust. The content of this file should be the CAs’ certificates encoded in PEM format. These trusted CA certificates are used by the driver during SSL handshake to verify the server certificate and determine if the server can be trusted. By default, the driver trusts the certificate authorities listed in the cacerts.pem file that comes with the driver.

Using Windows Azure HDInsight Emulator
This authentication mechanism is not supported in Cloudera's Distribution Including Apache Hadoop.

Using Windows Azure HDInsight Service
This authentication mechanism is not supported in Cloudera's Distribution Including Apache Hadoop.

Using HTTP
To configure HTTP authentication:
1. In the Cloudera Hive ODBC Driver DSN Setup dialog, click the drop-down arrow next to the Mechanism field in the Authentication area, and then select HTTP.
2. In the HTTP Path field, type the partial URL corresponding to the Hive server.

Note:
HTTP is only available starting with Hive 0.12.0.

Using HTTPS
To use HTTPS authentication:
1. In the Cloudera Hive ODBC Driver DSN Setup dialog, click the drop-down arrow next to the Mechanism field in the Authentication area, and then select HTTPS.
2. In the HTTP Path field, type the partial URL corresponding to the Hive server.
3. In the User Name field, type an appropriate user name for accessing the Hive server.
4. In the Password field, type the password corresponding to the user name you typed in step 3.

Note:
HTTPS is only available starting with Hive 0.13.0.

Note:
The driver always accepts the use of self-signed SSL certificate for this authentication mechanism.
**Note:**

Optionally, you can configure the **Allow Common Name Host Name Mismatch** setting to control whether the driver allows the common name of a CA issued certificate to not match the host name of the Hive server. For self-signed certificates, the driver always allows the common name of the certificate to not match the host name.

**Note:**

Optionally, you can configure the **Trusted Certificates** setting to specify the file listing the SSL certificate authorities (CAs) you would like the driver to trust. The content of this file should be the CAs’ certificates encoded in PEM format. These trusted CA certificates are used by the driver during SSL handshake to verify the server certificate and determine if the server can be trusted. By default, the driver trusts the certificate authorities listed in the cacerts.pem file that comes with the driver.

**Using Kerberos over HTTP**

To use Kerberos authentication, Kerberos must be configured prior to use. See “Appendix B: Configuring Kerberos Authentication for Windows” on page 34 for details.

After Kerberos has been installed and configured, then set the following options in the **Authentication** group in the Cloudera Hive ODBC Driver dialog box:

1. In the **Mechanism** field, select **Kerberos over HTTP**.
2. If there is no default realm configured for your Kerberos setup, then type the value for the Kerberos realm of the Hive Server 2 host. Otherwise leave it blank. The **Realm** is only needed if your Kerberos setup does not define a default realm or if the realm of your Hive Server 2 is not the default.
3. In the **Host FQDN** field, type the value for the fully-qualified domain name of the Hive Server 2 host.
4. In the **Service Name** field, type the value for the service name of the Hive Server 2. For example, if the principle for the Hive Server 2 is "hive/fully.qualified.domain.name@YOUR-REALM.COM", then the value in the service name field should be **hive**. If you are unsure of the correct service name to use for your particular Hadoop deployment, see your Hadoop administrator.
5. In the **HTTP Path** field, type the partial URL corresponding to the Hive server.

**Note:**

Kerberos over HTTP is only available starting with Hive 0.13.0.
Using Kerberos over HTTPS

To use Kerberos authentication, Kerberos must be configured prior to use. See “Appendix B: Configuring Kerberos Authentication for Windows” on page 34 for details.

After Kerberos has been installed and configured, then set the following options in the Authentication group in the Cloudera Hive ODBC Driver dialog box:

1. In the Mechanism field, select Kerberos over HTTPS.
2. If there is no default realm configured for your Kerberos setup, then type the value for the Kerberos realm of the Hive Server 2 host. Otherwise leave it blank. The Realm is only needed if your Kerberos setup does not define a default realm or if the realm of your Hive Server 2 is not the default.
3. In the Host FQDN field, type the value for the fully-qualified domain name of the Hive Server 2 host.
4. In the Service Name field, type the value for the service name of the Hive Server 2. For example, if the principle for the Hive Server 2 is "hive/fully.qualified.domain.name@YOUR-REALM.COM", then the value in the service name field should be hive. If you are unsure of the correct service name to use for your particular Hadoop deployment, see your Hadoop administrator.
5. In the HTTP Path field, type the partial URL corresponding to the Hive server.

Note:
Kerberos over HTTPS is only available starting with Hive 0.13.0.

Note:
The driver always accepts the use of self-signed SSL certificate for this authentication mechanism.

Note:
Optionally, you can configure the Allow Common Name Host Name Mismatch setting to control whether the driver allows the common name of a CA issued certificate to not match the host name of the Hive server. For self-signed certificates, the driver always allows the common name of the certificate to not match the host name.
Note:
Optionally, you can configure the **Trusted Certificates** setting to specify the file listing the SSL certificate authorities (CAs) you would like the driver to trust. The content of this file should be the CAs’ certificates encoded in PEM format. These trusted CA certificates are used by the driver during SSL handshake to verify the server certificate and determine if the server can be trusted. By default, the driver trusts the certificate authorities listed in the cacerts.pem file that comes with the driver.

Configuring DSN-Less Authentication

Some client applications, such as Tableau, provide some support for connecting to a data source using a driver without a DSN.

Applications that connect using ODBC data sources work with Hive Server 2 by sending the appropriate authentication credentials defined in the data source. Applications that are Hive Server 1 aware but not Hive Server 2 aware and that connect using a DSN-less connection will not have a facility for sending authentication credentials to Hive Server 2. However, you can configure Cloudera ODBC Driver for Hive with authentication credentials using the Driver Configuration tool.

**Important:**
Credentials defined in a data source take precedence over credentials configured using the Driver Configuration tool. Credentials configured using the Driver Configuration tool apply for all connections made using a DSN-less connection unless the client application is Hive Server 2 aware and requests credentials from the user.

To configure driver authentication for a DSN-less connection:

1. Click the **Start** button 📱.
2. Click **All Programs**.
3. Click the **Cloudera ODBC Driver for Apache Hive 2.5 (64-bit)** or the **Cloudera ODBC Driver for Apache Hive 2.5 (32-bit)** program group. If you installed both versions of the driver, you will see two program groups.

   Because drivers are bit-specific, select the version that matches the bitness of your application. For example, a 32-bit driver will only be accessible from 32-bit applications.

4. Click **Driver Configuration**, and then click **OK** if prompted for administrator permission to make modifications to the computer.

**Note:**
You must have administrator access to the computer in order to run this application because it makes changes to the registry.
5. Follow the procedure for configuring the Advanced Options on page 3 and the section “Configuring Authentication” on page 8 to complete the Cloudera Hive ODBC Driver Configuration dialog.

6. In the Cloudera Hive ODBC Driver Configuration dialog, click **OK**.

---

**Linux Driver**

**System Requirements**

- One of the following distributions (32- and 64-bit editions are supported):
  - Red Hat® Enterprise Linux® (RHEL) 5.0/6.0
  - CentOS 5.0/6.0
  - SUSE Linux Enterprise Server (SLES) 11

- 45 MB of available disk space.

- An installed ODBC driver manager:
  - iODBC 3.52.7 or above
    - OR
  - unixODBC 2.2.12 or above

Cloudera ODBC Driver for Hive requires a Hadoop cluster with the Hive service installed and running. Cloudera ODBC Driver for Hive is suitable for use with all versions of Hive.

**Installation**

There are two versions of the driver for Linux:

- **ClouderaHiveODBC-Version-Release.i686.rpm** for 32-bit
- **ClouderaHiveODBC-Version-Release.x86_64.rpm** for 64-bit

The version of the driver that you select should match the bitness of the client application accessing your Hadoop / Hive-based data. For example, if the client application is 64-bit, then you should install the 64-bit driver. Note that 64-bit editions of Linux support both 32- and 64-bit applications. Verify the bitness of your intended application and install the appropriate version of the driver.

**Important:**

Ensure that you install the driver using the RPM corresponding to your Linux distribution.

Cloudera ODBC Driver for Hive driver files are installed in the following directories:

- `/opt/cloudera/hiveodbc/ErrorMessages`—Error messages files directory
- `/opt/cloudera/hiveodbc/Setup`—Sample configuration files directory
- `/opt/cloudera/hiveodbc/lib/32`—32-bit shared libraries directory
Linux Driver

• /opt/cloudera/hiveodbc/lib/64—64-bit shared libraries directory

To install Cloudera ODBC Driver for Hive:

• In Red Hat Enterprise Linux 5.0/6.0 or CentOS 5.0/6.0, log in as the root user, then navigate to
the folder containing the driver RPM packages to install, and then type the following at the
command line, where RPMFileName is the file name of the RPM package containing the version
of the driver that you want to install:

```bash
yum --nogpgcheck localinstall RPMFileName
```

OR

In SUSE Linux Enterprise Server 11, log in as the root user, then navigate to the folder containing
the driver RPM packages to install, and then type the following at the command line, where
RPMFileName is the file name of the RPM package containing the version of the driver that you
want to install:

```bash
zypper install RPMFileName
```

Cloudera ODBC Driver for Hive depends on the following resources:

• cyrus-sasl-2.1.22-7 or above
• cyrus-sasl-gssapi-2.1.22-7 or above
• cyrus-sasl-plain-2.1.22-7 or above

If the package manager in your Linux distribution cannot resolve the dependencies automatically when
installing the driver, then download and manually install the packages required by the version of the
driver that you want to install.

Setting the LD_LIBRARY_PATH Environment Variable

The LD_LIBRARY_PATH environment variable must include the path to the installed ODBC driver
manager libraries.

For example, if you are using a 64-bit client application and ODBC driver manager libraries are installed
in /usr/local/lib, then set LD_LIBRARY_PATH as follows:

```bash
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
```

Refer to your Linux shell documentation for details on how to set environment variables permanently.

For details on creating ODBC connections using Cloudera ODBC Driver for Hive, see “Configuring ODBC
Connections for Linux, Mac OS X and AIX” on page 18.
Mac OS X Driver

System Requirements

- Mac OS X version 10.6.8 or later
- 100 MB of available disk space
- iODBC 3.52.7 or above

Cloudera ODBC Driver for Hive requires a Hadoop cluster with the Hive service installed and running.

Cloudera ODBC Driver for Hive is suitable for use with all versions of Hive. The driver supports both 32- and 64-bit client applications.

Installation

Cloudera ODBC Driver for Hive driver files are installed in the following directories:

- `/opt/cloudera/hiveodbc/ErrorMessages`—Error messages files directory
- `/opt/cloudera/hiveodbc/Setup`—Sample configuration files directory
- `/opt/cloudera/hiveodbc/lib/universal`—Binaries directory

To install Cloudera ODBC Driver for Hive:

1. Double-click to mount the `ClouderaHiveODBC.dmg` disk image.
2. Double-click `ClouderaHiveODBC.pkg` to run the Installer.
3. Follow the instructions in the Installer to complete the installation process.
4. When the installation completes, click Close.

Setting the DYLD_LIBRARY_PATH Environment Variable

The DYLD_LIBRARY_PATH environment variable must include the path to the installed ODBC driver manager libraries.

For example, if ODBC driver manager libraries are installed in `/usr/local/lib`, then set DYLD_LIBRARY_PATH as follows:

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

Refer to your Mac OS X shell documentation for details on how to set environment variables permanently.

For details on creating ODBC connections using Cloudera ODBC Driver for Hive, see “Configuring ODBC Connections for Linux, Mac OS X and AIX” on page 18.
AIX Driver

System Requirements

- IBM AIX 5.3, 6.1, or 7.1 (32- and 64-bit editions are supported)
- 150 MB of available disk space
- An installed ODBC driver manager:
  - iODBC 3.52.7 or above
  - OR
  - unixODBC 2.3.0 or above

Cloudera ODBC Driver for Hive requires a Hadoop cluster with the Hive service installed and running. The driver is suitable for use with all versions of Hive.

Installation

There are two versions of the driver for AIX:

- ClouderaHiveODBC-32bit-Version-Release.ppc.rpm for 32-bit
- ClouderaHiveODBC-Version-Release.ppc.rpm for 64-bit

The version of the driver that you select should match the bitness of the client application accessing your Hadoop / Hive-based data. For example, if the client application is 64-bit, then you should install the 64-bit driver. Note that 64-bit editions of AIX support both 32- and 64-bit applications. Verify the bitness of your intended application and install the appropriate version of the driver.

Cloudera ODBC Driver for Hive driver files are installed in the following directories:

- /opt/cloudera/hiveodbc/ErrorMessages—Error messages files directory
- /opt/cloudera/hiveodbc/Setup—Sample configuration files directory
- /opt/cloudera/hiveodbc/lib/32—32-bit shared libraries directory
- /opt/cloudera/hiveodbc/lib/64—64-bit shared libraries directory

To install Cloudera ODBC Driver for Hive:

1. Log in as root user, then navigate to the folder containing the driver RPM packages to install, and then type the following at the command line, where RPMFileName is the file name of the RPM package containing the version of the driver that you want to install:

```bash
rpm --install RPMFileName
```
Setting the LD_LIBRARY_PATH Environment Variable

The LD_LIBRARY_PATH environment variable must include the path to the installed ODBC driver manager libraries.

For details on creating ODBC connections using Cloudera ODBC Driver for Hive, see “Configuring ODBC Connections for Linux, Mac OS X and AIX” on page 18.

Configuring ODBC Connections for Linux, Mac OS X and AIX

Files

ODBC driver managers use configuration files to define and configure ODBC data sources and drivers. By default, the following configuration files residing in the user’s home directory are used:

- .odbc.ini – The file used to define ODBC data sources (required)
- .odbcinst.ini – The file used to define ODBC drivers (optional)

By default, the cloudera.hiveodbc.ini in the /opt/cloudera/hiveodbc/lib/32 directory for 32-bit AIX/Linux drivers, /opt/cloudera/hiveodbc/lib/64 directory for 64-bit AIX/Linux drivers or /opt/cloudera/hiveodbc/lib/universal directory for Mac OS X driver is used to configure Cloudera ODBC Driver for Hive (required).

Sample Files

The driver installation contains the following sample configuration files in the Setup directory:

- odbc.ini
- odbcinst.ini

The names of the sample configuration files do not begin with a period (.) so that they will appear in directory listings by default. A filename beginning with a period (.) is hidden. For odbc.ini and odbcinst.ini, if the default location is used, then the filenames must begin with a period (.).

If the configuration files do not already exist in the user’s home directory, then the sample configuration files can be copied to that directory and renamed. If the configuration files already exist in the user’s home directory, then the sample configuration files should be used as a guide for modifying the existing configuration files.

Configuring the Environment

By default, the configuration files reside in the user’s home directory. However, two environment variables, ODBCINI and ODBCSYSINI, can be used to specify different locations for the odbc.ini and odbcinst.ini configuration files. Set ODBCINI to point to your odbc.ini file. Set ODBCSYSINI to point to the directory containing the odbcinst.ini file.

The driver’s configuration file (cloudera.hiveodbc.ini) is installed to the driver’s lib directory and used by the driver by default. The CLOUDERAHIVEINI environment variable can be used to specify a different location for the cloudera.hiveodbc.ini file.
Configuring ODBC Connections for Linux, Mac OS X and AIX

For example, if your odbc.ini and cloudera.hiveodbc.ini files are located in /etc and your odbcinst.ini file is located in /usr/local/odbc, then set the environment variables as follows:

```bash
export ODBCINI=/etc/odbc.ini
export ODBCSYSINI=/usr/local/odbc
export CLOUDERAHIVEINI=/etc/cloudera.hiveodbc.ini
```

The search order for the cloudera.hiveodbc.ini file is as follows:

1. If the CLOUDERAHIVEINI environment variable is defined, then the driver searches for the file specified by the environment variable.

   **Important:**
   CLOUDERAHIVEINI must specify the full path, including the filename.

2. The directory containing the driver’s binary is searched for a file named cloudera.hiveodbc.ini **not** beginning with a period.

3. The current working directory of the application is searched for a file named cloudera.hiveodbc.ini **not** beginning with a period.

4. The directory `~/` (i.e. `$HOME`) is searched for a hidden file named `.cloudera.hiveodbc.ini`

5. The directory `/etc` is searched for a file named cloudera.hiveodbc.ini **not** beginning with a period.

**Configuring the odbc.ini File**

ODBC Data Sources are defined in the odbc.ini configuration file. The file is divided into several sections:

- **[ODBC]** is optional and used to control global ODBC configuration, such as ODBC tracing.
- **[ODBC Data Sources]** is required, listing DSNs and associating DSNs with a driver.
- A section having the same name as the data source specified in the [ODBC Data Sources] section is required to configure the data source.

Here is an example odbc.ini configuration file for Linux or AIX:

```ini
[ODBC Data Sources]
Sample Cloudera Hive DSN 32=Cloudera Hive ODBC Driver 32-bit
[Sample Cloudera Hive DSN 32]
Driver=/opt/cloudera/hiveodbc/lib/32/libclouderahiveodbc32.so
HOST=MyHiveServer
PORT=10000
```
Configuring ODBC Connections for Linux, Mac OS X and AIX

**Note:**
To connect to Hive Server 2 you must add `HiveServerType=2` in the DSN.

Here is an example odbc.ini configuration file for Mac OS X:

```
[ODBC Data Sources]
Sample Cloudera Hive DSN=Cloudera Hive ODBC Driver
[Sample Cloudera Hive DSN]
Driver=/opt/cloudera/hiveodbc/lib/universal/libclouderahiveodbc.dylib
HOST=MyHiveServer
PORT=10000
```

**Note:**
To connect to Hive Server 2 you must add `HiveServerType=2` in the DSN.

To create a data source:

1. Open the `.odbc.ini` configuration file in a text editor.
2. Add a new entry to the [ODBC Data Sources] section. Type the data source name (DSN) and the driver name.
3. To set configuration options, add a new section that has a name that matches the data source name (DSN) you specified in step 2. Specify configuration options as key-value pairs.
4. Save the `.odbc.ini` configuration file.

For details on configuration options available to control the behavior of DSNs using Cloudera ODBC Driver for Hive, see “Appendix C: Driver Configuration Options” on page 38.

**Configuring the odbcinst.ini File**

ODBC Drivers are defined in the odbcinst.ini configuration file. The configuration file is optional because drivers can be specified directly in the odbc.ini configuration file, as described in “Configuring the odbc.ini File” on page 19.

The odbcinst.ini file is divided into the following sections:

- **[ODBC Drivers]** lists the names of all the installed ODBC drivers.
- A section having the same name as the driver name specified in the [ODBC Drivers] section lists driver attributes and values.
Configuring ODBC Connections for Linux, Mac OS X and AIX

Here is an example odbcinst.ini file for Linux or AIX:

```
[ODBC Drivers]
Cloudera Hive ODBC Driver 32-bit=Installed
Cloudera Hive ODBC Driver 64-bit=Installed
[Cloudera Hive ODBC Driver 32-bit]
Description=Cloudera Hive ODBC Driver (32-bit)
Driver=/opt/cloudera/hiveodbc/lib/32/libclouderahiveodbc32.so
[Cloudera Hive ODBC Driver 64-bit]
Description=Cloudera Hive ODBC Driver (64-bit)
Driver=/opt/cloudera/hiveodbc/lib/64/libclouderahiveodbc64.so
```

Here is an example odbcinst.ini file for Mac OS X:

```
[ODBC Drivers]
Cloudera Hive ODBC Driver=Installed
[Cloudera Hive ODBC Driver]
Description=Cloudera Hive ODBC Driver
Driver=/opt/cloudera/hiveodbc/lib/universal/libclouderahiveodbc.dylib
```

To define a driver:

1. Open the .odbcinst.ini configuration file in a text editor.
2. Add a new entry to the [ODBC Drivers] section. Type the driver name, and then type the following: =Installed

   **Note:**
   
   Assign the driver name as the value of the Driver attribute in the data source definition instead of the driver shared library name.

3. In .odbcinst.ini, add a new section that has a name that matches the driver name you typed in step 2, and then add configuration options to the section based on the sample odbcinst.ini file provided with Cloudera ODBC Driver for Hive in the Setup directory. Specify configuration options as key-value pairs.
4. Save the .odbcinst.ini configuration file.

**Configuring the cloudera.hiveodbc.ini File**

To configure Cloudera ODBC Driver for Hive to work with your ODBC driver manager:

1. Open the .cloudera.hiveodbc.ini configuration file in a text editor.
2. Edit the DriverManagerEncoding setting. If you are using Linux or Mac OS X, the value usually must be UTF-16 or UTF-32 depending on the ODBC driver manager you use. iODBC uses UTF-32 and unixODBC uses UTF-16. Consult your ODBC Driver Manager documentation for the correct setting to use.

OR

If you are using AIX, then set the value to UTF-16 if you are using the unixODBC driver manager. If you are using the iODBC driver manager, then set the value to UTF-16 if you are using the 32-bit driver, or set the value to UTF-32 if you are using the 64-bit driver.

3. Edit the ODBCInstLib setting. The value is the name of the ODBCInst shared library for the ODBC driver manager you use. The configuration file defaults to the shared library for iODBC. In Linux, the shared library name for iODBC is libiodbcinst.so. In Mac OS X, the shared library name for iODBC is libiodbcinst.dylib.

Note:
Consult your ODBC driver manager documentation for the correct library to specify. You can specify an absolute or relative filename for the library. If you intend to use the relative filename, then the path to the library must be included in the library path environment variable. In Linux, the library path environment variable is named LD_LIBRARY_PATH. In Mac OS X, the library path environment variable is named DYLD_LIBRARY_PATH.

4. Save the .cloudera.hiveodbc.ini configuration file.

Configuring Authentication

For details on selecting the appropriate authentication for a DSN using Cloudera ODBC Driver for Hive, see “Appendix A: Authentication Options” on page 30.

For details on the keys involved in configuring authentication, see “Appendix A: Authentication Options” on page 30. The authentication methods available are as follows:

- No Authentication
- Kerberos
- User Name
- User Name and Password
- User Name and Password (SSL)
- HTTP
- HTTPS
- Kerberos over HTTP
- Kerberos over HTTPS

Using No Authentication

No additional details are required when using No Authentication.
Using Kerberos

For information on operating Kerberos, refer to the documentation for your operating system.

**To configure a DSN using Cloudera ODBC Driver for Hive to use Kerberos authentication:**

1. Set the AuthMech configuration key for the DSN to 1.
2. If your Kerberos setup does not define a default realm or if the realm of your Hive server is not the default, then set the appropriate realm using the KrbRealm key.
3. Set the KrbHostFQDN key to the fully qualified domain name of the Hive Server 2 host.
4. Set the KrbServiceName key to the service name of the Hive Server 2. For example, if the principle for the Hive Server 2 is "hive/fully.qualified.domain.name@YOUR-REALM.COM", then the value in the service name field should be hive. If you are unsure of the correct service name to use for your particular Hadoop deployment, see your Hadoop administrator.

Using User Name

**To configure User Name authentication:**

1. Set the AuthMech configuration key for the DSN to 2.
2. Set the UserName key to the appropriate credential recognized by the Hive server.

Using User Name and Password

**To configure User Name and Password authentication:**

1. Set the AuthMech configuration key for the DSN to 3.
2. Set the UID key to the appropriate user name recognized by the Hive server.
3. Set the PWD key to the password corresponding to the user name you provided in step 2.

Using User Name and Password (SSL)

**To configure User Name and Password (SSL) authentication:**

1. Set the AuthMech configuration key for the DSN to 4.
2. Set the UID key to the appropriate user name recognized by the Hive server.
3. Set the PWD key to the password corresponding to the user name you provided in step 2.

**Note:**

The driver always accepts the use of self-signed SSL certificate for this authentication mechanism.
Configuring ODBC Connections for Linux, Mac OS X and AIX

Note:
Optionally, you can configure the **CAIssuedCertNamesMismatch** setting to control whether the driver allows the common name of a CA-issued certificate to not match the host name of the Hive server. For self-signed certificates the driver always allows the common name of the certificate to not match the host name. See “Appendix C: Driver Configuration Options” on page 38.

Note:
Optionally, you can configure the **TrustedCerts** setting to specify the file listing the SSL certificate authorities (CAs) you would like the driver to trust. The content of this file should be the CAs’ certificates encoded in PEM format. These trusted CA certificates are used by the driver during SSL handshake to verify the server certificate and determine if the server can be trusted. By default the driver trusts the certificate authorities listed in the cacerts.pem file that comes with the driver. See “Appendix C: Driver Configuration Options” on page 38.

Using HTTP

To configure HTTP authentication:
1. Set the AuthMech configuration key for the DSN to 7.
2. Set the HTTPPath key to the partial URL corresponding to the Hive server.

Note:
HTTP is only available starting with Hive 0.12.0.

Using HTTPS

To configure HTTPS authentication:
1. Set the AuthMech configuration key for the DSN to 8.
2. Set the HTTPPath key to the partial URL corresponding to the Hive server.
3. Set the UID key to an appropriate user name for accessing the Hive server.
4. Set the PWD key to the password corresponding to the user name you typed in step 3.

Note:
HTTPS is only available starting with Hive 0.13.0.
Note:
The driver always accepts the use of self-signed SSL certificate for this authentication mechanism.

Note:
Optionally, you can configure the `CAIssuedCertNamesMismatch` setting to control whether the driver allows the common name of a CA-issued certificate to not match the host name of the Hive server. For self-signed certificates, the driver always allows the common name of the certificate to not match the host name. See “Appendix C: Driver Configuration Options” on page 38.

Note:
Optionally, you can configure the `TrustedCerts` setting to specify the file listing the SSL certificate authorities (CAs) you would like the driver to trust. The content of this file should be the CAs’ certificates encoded in PEM format. These trusted CA certificates are used by the driver during SSL handshake to verify the server certificate and determine if the server can be trusted. By default the driver trusts the certificate authorities listed in the cacerts.pem file that comes with the driver. See “Appendix C: Driver Configuration Options” on page 38.

Using Kerberos over HTTP
For information on operating Kerberos, refer to the documentation for your operating system.

To configure a DSN using Cloudera ODBC Driver for Hive to use Kerberos over HTTP authentication:

1. Set the `AuthMech` configuration key for the DSN to 9.
2. If your Kerberos setup does not define a default realm or if the realm of your Hive server is not the default, then set the appropriate realm using the `KrbRealm` key.
3. Set the `KrbHostFQDN` key to the fully qualified domain name of the Hive Server 2 host.
4. Set the `KrbServiceName` key to the service name of the Hive Server 2. For example, if the principle for the Hive Server 2 is “hive/fully.qualified.domain.name@YOUR-REALM.COM”, then the value in the service name field should be `hive`. If you are unsure of the correct service name to use for your particular Hadoop deployment, see your Hadoop administrator.
5. Set the `HTTPPath` key to the partial URL corresponding to the Hive server.

Note:
Kerberos over HTTP is only available starting with Hive 0.13.0.

Using Kerberos over HTTPS
For information on operating Kerberos, refer to the documentation for your operating system.

To configure a DSN using Cloudera ODBC Driver for Hive to use Kerberos over HTTPS authentication:

1. Set the `AuthMech` configuration key for the DSN to 10.
2. If your Kerberos setup does not define a default realm or if the realm of your Hive server is not the default, then set the appropriate realm using the KrbRealm key.

3. Set the KrbHostFQDN key to the fully qualified domain name of the Hive Server 2 host.

4. Set the KrbServiceName key to the service name of the Hive Server 2. For example, if the principle for the Hive Server 2 is "hive/fully.qualified.domain.name@YOUR-REALM.COM", then the value in the service name field should be hive. If you are unsure of the correct service name to use for your particular Hadoop deployment, see your Hadoop administrator.

5. Set the HTTPPath key to the partial URL corresponding to the Hive server.

**Note:**

Kerberos over HTTPS is only available starting with Hive 0.13.0.

**Note:**

The driver always accepts the use of self-signed SSL certificate for this authentication mechanism.

**Note:**

Optionally, you can configure the CAIssuedCertNamesMismatch setting to control whether the driver allows the common name of a CA-issued certificate to not match the host name of the Hive server. For self-signed certificates the driver always allows the common name of the certificate to not match the host name. See “Appendix C: Driver Configuration Options” on page 38.

**Note:**

Optionally, you can configure the TrustedCerts setting to specify the file listing the SSL certificate authorities (CAs) you would like the driver to trust. The content of this file should be the CAs’ certificates encoded in PEM format. These trusted CA certificates are used by the driver during SSL handshake to verify the server certificate and determine if the server can be trusted. By default the driver trusts the certificate authorities listed in the cacerts.pem file that comes with the driver. See “Appendix C: Driver Configuration Options” on page 38.

---

**Features**

### SQL Query versus HiveQL Query

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

**SQL Connector**

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

- **Quoted Identifiers**—When quoting identifiers, HiveQL uses back quotes (`) while SQL uses double quotes (""). Even when a driver reports the back quote as the quote character, some applications still generate double-quoted identifiers.

- **Table Aliases**—HiveQL does not support the AS keyword between a table reference and its alias.

- **JOIN, INNER JOIN and CROSS JOIN**—SQL INNER JOIN and CROSS JOIN syntax is translated to HiveQL JOIN syntax.

- **TOP N/LIMIT**—SQL TOP N queries are transformed to HiveQL LIMIT queries.

**Data Types**

The following data types are supported:

- TINYINT
- SMALLINT
- INT
- BIGINT
- FLOAT
- DOUBLE
- DECIMAL
- BOOLEAN
- STRING
- TIMESTAMP
- VARCHAR(n)
- DATE
- DECIMAL(p,s)
- CHAR(n)

**Note:**

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

Catalog and Schema Support

Cloudera ODBC Driver for Hive supports both catalogs and schemas in order to make it easy for the driver to work with various ODBC applications. Since Hive only organizes tables into schema/database, we have added a synthetic catalog, called “HIVE” under which all of the schemas/databases are organized. The driver also maps the ODBC schema to the Hive schema/database.

Hive System Table

A pseudo-table called HIVE_SYSTEM can be used to query for Hive cluster system environment information. The pseudo table is under the pseudo schema HIVE_SYSTEM. The table has two String type columns ENVKEY and ENVVALUE. Standard SQL can be executed against the Hive system table. For example:

```
SELECT * FROM HIVE_SYSTEM.HIVE_SYSTEM WHERE ENVKEY LIKE '%hive%'
```

The example query returns all of the Hive system environment entries whose key contains the word “hive.” A special query, “set -v”, is executed to fetch system environment information and is not supported by all Hive versions. For versions of Hive that do not support querying system environment information, the driver returns an empty result set.

Server-side Properties

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

For details on setting server-side properties for a DSN using the Windows driver, see Configuring ODBC Connections on page 2. For details related to the Linux and Mac OS X drivers, see “Appendix C: Driver Configuration Options” on page 38.

Active Directory

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

1. MIT Kerberos is not installed on client Windows machine.
2. The MIT Kerberos Hadoop realm has been configured to trust the Active Directory realm, according to Cloudera’s documentation, so that users in the Active Directory realm can access services in the MIT Kerberos Hadoop realm.

Get Tables With Query

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

Hive Server 2 has a limit on the number of tables that can be in a database when handling the GetTables API call. When the number of tables in a database is above the limit, the API call will return a stack overflow error or a timeout error. The exact limit and the error that appears depend on the JVM settings.
As a workaround for this issue, enable the **Get Tables with Query** (or **GetTablesWithQuery**)
configuration option to use the query instead of the API call.

**Temporary Table**

The Temporary Table feature adds support for creating temporary tables and inserting literal values into
temporary tables. Temporary tables are only accessible by the ODBC connection that created them and
they will be dropped upon disconnect. For temporary table CREATE TABLE and INSERT statement syntax
see “Appendix D: Temporary Table CREATE TABLE and INSERT Statements” on page 49.

**Contact Us**

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

**Important:**

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

For details on contacting Technical Support, see
http://www.cloudera.com/content/cloudera/en/products/cloudera-support.html
Appendix A: Authentication Options

**Note:**

Authentication is only available for servers of type Hive Server 2. Authentication is not available for servers of type Hive Server 1.

Hive Server 2 supports multiple authentication mechanisms. You must determine the authentication type your server is using. The authentication methods available are as follows:

- No Authentication
- Kerberos
- User Name
- User Name and Password
- User Name and Password (SSL)
- HTTP
- HTTPS
- Kerberos over HTTP
- Kerberos over HTTPS

To discover how your Hive Server 2 is configured, examine your hive-site.xml file. Examine the following properties to determine which authentication mechanism your server is set to use:

- hive.server2.authentication
- hive.server2.enable.doAs

<table>
<thead>
<tr>
<th>hive.server2.authentication</th>
<th>hive.server2.enable.doAs</th>
<th>Driver Authentication Mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOSASL</td>
<td>False</td>
<td>No Authentication</td>
</tr>
<tr>
<td>KERBEROS</td>
<td>True or False</td>
<td>Kerberos</td>
</tr>
<tr>
<td>NONE</td>
<td>True or False</td>
<td>User Name</td>
</tr>
</tbody>
</table>
| LDAP                        | True or False            | User Name and Password OR User Name and Password (SSL)  
**Note:** User Name and Password (SSL) can only be used if your Hive server is configured with SSL.
Hive Server 2 uses SASL (Simple Authentication and Security Layer) to support some of the authentication methods. The following table shows which authentication methods are supported by SASL.

<table>
<thead>
<tr>
<th>SASL mechanisms</th>
<th>Non-SASL mechanisms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kerberos</td>
<td>No Authentication</td>
</tr>
<tr>
<td>User Name</td>
<td>HTTP</td>
</tr>
<tr>
<td>User Name and Password</td>
<td>HTTPS</td>
</tr>
<tr>
<td>User Name and Password (SSL)</td>
<td>Kerberos over HTTP</td>
</tr>
<tr>
<td></td>
<td>Kerberos over HTTPS</td>
</tr>
</tbody>
</table>

*Kerberos* is supported by the SASL GSSAPI mechanism, while *User Name*, *User Name and Password*, and *User Name and Password (SSL)* are supported by the SASL PLAIN mechanism.

**Note:**
Remote process communication between the Cloudera ODBC Driver for Hive and the Hive Server is handled by a layer called Thrift. Thrift cannot detect combinations of both non-SASL and SASL mechanisms being used between the driver and the server. For example, if the driver is using a non-SASL mechanism but the server is using a SASL mechanism, then the driver will hang when attempting to establish a connection. Ensure that the driver and the server both use mechanisms of the same type.

**Note:**
The default configuration of Hive Server 2 uses the SASL PLAIN mechanism and requires you to use either the *User Name* or *User Name and Password* authentication mechanism for the Cloudera ODBC Driver for Hive.

For more detail on authentication mechanisms, see the documentation for your Hadoop / Hive distribution. See also “Running Hadoop in Secure Mode” at
Appendix A: Authentication Options

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 hive.server2.authentication is set to KERBEROS, then you must configure your connection to use Kerberos.

Using User Name
When hive.server2.authentication is set to NONE, you must configure your connection to use User Name. 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 DSN or driver configuration must be an existing OS user on the host running Hive Server 2.
- If hive.server2.enable.doAs is set to false, then the User Name in the DSN or driver configuration is ignored.

If the User Name in the DSN or driver configuration is not supplied, then the driver defaults to using “anonymous” as the user name.

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

Using User Name and Password (SSL)
When connecting to a Hive server of type Hive Server 2 configured to use SASL-PLAIN authentication with user name and password, and with SSL enabled, then you must configure your connection to use User Name and Password.

Using HTTP
When connecting to Hive Server 2 configured to use the Thrift HTTP transport over a TCP socket then you must configure your connection to use HTTP authentication mechanism.

Using HTTPS
When connecting to Hive Server 2 configured to use the Thrift HTTP transport over a SSL socket then you must configure your connection to use HTTPS authentication mechanism.

Using Kerberos over HTTP
When connecting to a Kerberos enabled Hive Server 2 configured to use the Thrift HTTP transport over a TCP socket, you must configure your connection to use Kerberos over HTTP authentication mechanism.
Using Kerberos over HTTPS

When connecting to Kerberos enabled Hive Server 2 configured to use the Thrift HTTP transport over a SSL socket, you must configure your connection to use Kerberos over HTTPS authentication mechanism.
Appendix B: Configuring Kerberos Authentication for Windows

Active Directory

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

1. MIT Kerberos is not installed on client Windows machine.
2. The MIT Kerberos Hadoop realm has been configured to trust the Active Directory realm, according to Cloudera’s documentation, so that users in the Active Directory realm can access services in the MIT Kerberos Hadoop realm.

MIT Kerberos

Download and install MIT Kerberos for Windows 4.0.1

1. For 64-bit computers: [http://web.mit.edu/kerberos/dist/kfw/4.0/kfw-4.0.1-amd64.msi](http://web.mit.edu/kerberos/dist/kfw/4.0/kfw-4.0.1-amd64.msi). The installer includes both 32-bit and 64-bit libraries.
2. For 32-bit computers: [http://web.mit.edu/kerberos/dist/kfw/4.0/kfw-4.0.1-i386.msi](http://web.mit.edu/kerberos/dist/kfw/4.0/kfw-4.0.1-i386.msi). The installer includes 32-bit libraries only.

Set up the Kerberos configuration file in the default location

1. Obtain a krb5.conf configuration file from your Kerberos administrator. The configuration file should also be present at /etc/krb5.conf on the machine hosting the Hive Server 2.
2. The default location is C:\ProgramData\MIT\Kerberos5 but this is normally a hidden directory. Consult your Windows documentation if you wish to view and use this hidden directory.
3. Rename the configuration file from krb5.conf to krb5.ini.
4. Copy krb5.ini to the default location and overwrite the empty sample file.

Consult the MIT Kerberos documentation for more information on configuration.

Set up the Kerberos configuration file in another location

If you do not want to put the Kerberos configuration file in the default location then you can use another location. The steps required to do this are as follows:

1. Obtain a krb5.conf configuration file for your Kerberos setup.
2. Store krb5.conf in an accessible directory and make note of the full path name.
3. Click the Windows Start menu.
4. Right-click Computer.
5. Click Properties.
6. Click Advanced system settings.
7. Click Environment Variables.
8. Click New for System variables.
9. In the Variable Name field, type `KRBS_CONFIG`.
10. In the Variable Value field, type the absolute path to the krb5.conf file you stored in step 2.
11. Click OK to save the new variable.
12. Ensure the variable is listed in the System variables list.
13. Click OK to close Environment Variables Window.
14. Click OK to close System Properties Window.

Set up the Kerberos credential cache file

1. Create a new directory where you want to save the Kerberos credential cache file. For example, you may create the C:\temp directory.
2. Click the Windows Start menu.
3. Right-click Computer.
4. Click Properties.
5. Click Advanced system settings.
6. Click Environment Variables.
7. Click New for System variables.
8. In the Variable Name field, type `KRBSCCNAME`.
9. In the Variable Value field, type the path to the directory you created in step 1, and then append the file name `krb5cache`. For example, if you created the C:\temp directory in step 1, then type: C:\temp\krb5cache

**Note:**

`krb5cache` is a file—not a directory—managed by the Kerberos software and should not be created by the user. If you receive a permission error when you first use Kerberos, check to ensure that the krb5cache file does not exist as a file or a directory.

10. Click OK to save the new variable.
11. Ensure the variable appears in the System variables list.
12. Click OK to close Environment Variables Window.
13. Click OK to close System Properties Window.
14. Restart your computer to ensure that MIT Kerberos for Windows uses the new settings.

Obtain a ticket for a Kerberos principal using password

**Note:**

If your Kerberos environment uses keytab files please see the next section.

1. Click the Start button 🕒.
Appendix B: Configuring Kerberos Authentication for Windows

2. Click All Programs.
3. Click the Kerberos for Windows (64-bit) or the Kerberos for Windows (32-bit) program group.
4. Use MIT Kerberos Ticket Manager to obtain a ticket for the principal that will be connecting to Hive Server 2.

Obtain a ticket for a Kerberos principal using a keytab file

1. Click the Start button .
2. Click All Programs.
3. Click Accessories.
4. Click Command Prompt.
5. Type: kinit -k -t <keytab pathname> <principal>

   <keytab pathname> is the full pathname to the keytab file. For example, C:\mykeytabs\hiveserver2.keytab

   <principal> is the Kerberos principal to use for authentication. For example, hive/hiveserver2.example.com@EXAMPLE.COM

On some Windows systems, the cache location KRB5CCNAME is either not set or not used. In this case, use the '-c' option of the kinit command to specify the proper ticket cache that needs to be populated.

The command syntax is:

```
kinit -k -t C:\mykeytabs\hive.keytab hive/HOST@HADOOP.NET -c c:\ProgramData\MIT\krbcache
```

(krbcache is the kerberos cache file, not a directory.)

**Note:**
The order of the options shown above is important because the '-c' argument must be last.

Obtain a ticket for a Kerberos principal using the default keytab file

A default keytab file can be set for your Kerberos configuration. Consult the MIT Kerberos documentation for instructions on configuring a default keytab file.

1. Click the Start button .
2. Click All Programs.
3. Click Accessories.
4. Click Command Prompt.
5. **Type:** `kinit -k <principal>`

`<principal>` is the Kerberos principal to use for authentication. For example, `hive/hiveserver2.example.com@EXAMPLE.COM`

On some Windows systems, the cache location KRB5CCNAME is either not set or not used. In this case, use the `-c` option of the `kinit` command to specify the proper ticket cache that needs to be populated.

The command syntax is:

```
kinit -k hive/HOST@HADOOP.NET -c c:\ProgramData\MIT\krbcache
```

(krbcache is the kerberos cache file, not a directory.)

**Note:**

The order of the options shown above is important because the `-c` argument must be last.
Appendix C: Driver Configuration Options

The configuration options available to control the behavior of Cloudera ODBC Driver for Hive are listed and described in Table 1.

**Note:**
You can set configuration options in your odbc.ini and cloudera.hiveodbc.ini files. Configuration options set in a cloudera.hiveodbc.ini file apply to all connections, whereas configuration options set in an odbc.ini file are specific to a connection. Configuration options set in odbc.ini take precedence over configuration options set in cloudera.hiveodbc.ini.

### Table 1 Driver Configuration Options

<table>
<thead>
<tr>
<th>Key</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driver</td>
<td></td>
<td>The location of the Cloudera ODBC Driver for Hive shared object file.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Required)</td>
</tr>
<tr>
<td>HOST</td>
<td></td>
<td>The IP address or hostname of the Hive server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Required)</td>
</tr>
<tr>
<td>PORT</td>
<td>10000</td>
<td>The listening port for the service.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Required)</td>
</tr>
<tr>
<td>Schema</td>
<td>default</td>
<td>The name of the database schema to use when a schema is not explicitly specified in a query.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Optional)</td>
</tr>
</tbody>
</table>

**Note:**
Queries on other schemas can still be issued by explicitly specifying the schema in the query. To determine the appropriate database schema to use, type `show databases` at the Hive command prompt to inspect your databases.
## Appendix C: Driver Configuration Options

<table>
<thead>
<tr>
<th>Key</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DefaultStringColumnLength</td>
<td>255</td>
<td>The default string column length to use. Hive does not provide the length for String columns in its column metadata. This option allows you to tune the length of String columns. (Optional)</td>
</tr>
<tr>
<td>BinaryColumnLength</td>
<td>32767</td>
<td>The maximum data length for binary columns.</td>
</tr>
<tr>
<td>Note:</td>
<td></td>
<td><strong>Hive does not provide the maximum data length for binary columns in the columns metadata. The option allows you to tune the maximum data length for binary columns.</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Optional)</td>
</tr>
<tr>
<td>UseNativeQuery</td>
<td>0</td>
<td>Enabling the UseNativeQuery option using a value of 1 disables the SQL Connector feature. The SQL Connector feature allows the driver to apply transformations to the queries emitted by an application to convert them into an equivalent form in HiveQL. If the application is Hive-aware and already emits HiveQL, then turning off the SQL Connector feature avoids the extra overhead of query transformation. (Optional)</td>
</tr>
</tbody>
</table>
### Appendix C: Driver Configuration Options

<table>
<thead>
<tr>
<th>Key</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FastSQLPrepare</td>
<td>0</td>
<td>To enable the FastSQLPrepare option, use a value of 1. Enabling FastSQLPrepare defers query execution to SQLExecute. When using Native Query mode, the driver will execute the HiveQL query to retrieve the result set metadata for SQLPrepare. As a result, SQLPrepare might be slow. If the result set metadata is not required after calling SQLPrepare, then enable FastSQLPrepare. (Optional)</td>
</tr>
<tr>
<td>EnableAsyncExec</td>
<td>0</td>
<td>Enable/disable the use of asynchronous query execution. Note: This option only takes effect when connecting to Hive cluster running Hive 0.12.0 or higher.</td>
</tr>
</tbody>
</table>

**Note:**
Due to a problem in Hive 0.12.0 (see [https://issues.apache.org/jira/browse/HIVE-5230](https://issues.apache.org/jira/browse/HIVE-5230)), Hive returns generic error messages for errors that occur during query execution. To see the actual error message relevant to the problem, turn off asynchronous query execution and execute the query again.

Set to 1 to enable.
Set to 0 to disable.
(Optional)
<table>
<thead>
<tr>
<th>Key</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RowsFetchedPerBlock</td>
<td>10000</td>
<td>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. (Optional)</td>
</tr>
<tr>
<td>DecimalColumnScale</td>
<td>10</td>
<td>The maximum number of digits to the right of the decimal point for numeric data types. (Optional)</td>
</tr>
<tr>
<td>SSP</td>
<td></td>
<td>To set a server-side property, use the following syntax where SSPKey is the name of the server-side property to set and SSPValue is the value to assign to the server-side property: SSP_SSPKey=SSPValue For example: SSP_mapred.queue.names=myQueue After the driver applies the server-side property, the SSP_ prefix is removed from the DSN entry leaving an entry of SSPKey=SSPValue Important: The SSP_ prefix is case sensitive. (Optional)</td>
</tr>
</tbody>
</table>
### Appendix C: Driver Configuration Options

<table>
<thead>
<tr>
<th>Key</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ApplySSPWithQueries</strong></td>
<td>1</td>
<td>When set to the default value of 1 (enabled), each server-side property you set is applied by executing a set SSPKey=SSPValue query when opening a session to the Hive server. Applying server-side properties using queries involves an additional network round trip per server-side property when establishing a session to the Hive server. Some Hive Server 2 builds are not compatible with the more efficient method of setting server-side properties that the driver uses when ApplySSPWithQueries is disabled by this to 0.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td></td>
<td>When connecting to a Hive Server 1, ApplySSPWithQueries is always enabled. Set to 1 to enable. Set to 0 to disable. (Optional)</td>
</tr>
<tr>
<td><strong>LCaseSspKeyName</strong></td>
<td>1</td>
<td>Controls whether the driver will convert server-side property key name to all lower case characters. Set to 1 to enable. Set to 0 to disable. (Optional)</td>
</tr>
<tr>
<td><strong>HiveServerType</strong></td>
<td>1</td>
<td>The Hive Server Type. Set to 1 for Hive Server 1. Set to 2 for Hive Server 2. (Optional)</td>
</tr>
</tbody>
</table>
## Appendix C: Driver Configuration Options

<table>
<thead>
<tr>
<th>Key</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AuthMech</td>
<td>0</td>
<td>The authentication mechanism to use. Set the value to one of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0 for no authentication</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 1 for Kerberos</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 2 for User Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 3 for User Name and Password</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 4 for User Name and Password (SSL)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 5 (Reserved)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 6 (Reserved)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 7 for HTTP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 8 for HTTPS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 9 for Kerberos over HTTP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 10 for Kerberos over HTTPS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Optional)</td>
</tr>
<tr>
<td>KrbHostFQDN</td>
<td></td>
<td>The fully qualified domain name of the Hive Server 2 host used. (Required if AuthMech is Kerberos)</td>
</tr>
<tr>
<td>KrbServiceName</td>
<td></td>
<td>The Kerberos service principal name of the Hive Server 2. By convention the service name is hive, but the name may be different in your server environment. (Required if AuthMech is Kerberos)</td>
</tr>
</tbody>
</table>
## Appendix C: Driver Configuration Options

<table>
<thead>
<tr>
<th>Key</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| KrbRealm                     |               | If there is no default realm configured or the realm of the Hive Server 2 host is different from the default realm for your Kerberos setup, then define the realm of the Hive Server 2 host using this option.  
(Optional)                   |
| HTTPPath                     |               | The partial URL corresponding to Hive server configured to use HTTP or HTTPS.  
(Optional)                   |
| UID                          | anonymous     | The user name of the user credential.  
(Required if AuthMech is User Name and Password, User Name and Password (SSL), or HTTPS)  
(Optional if AuthMech is User Name) |
| PWD                          |               | The password of the user credential.  
(Required if AuthMech is User Name and Password, User Name and Password (SSL), or HTTPS) |
| CAIssuedCertNamesMismatch    | 0             | Control whether to allow CA issued SSL certificate’s common name to not match the host name of the Hive server.  
Set to 1 to enable.  
Set to 0 to disable.  

**Note:**  
This setting is only applicable to User Name and Password (SSL) authentication mechanism and will be ignored by other authentication mechanisms.  
(Optional) |
<table>
<thead>
<tr>
<th>Key</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| TrustedCerts           | For 32 bit driver: /opt/cloudera/hiveodbc/lib/32/ca certs.pem  
For 64 bit driver: /opt/cloudera/hiveodbc/lib/64/ca certs.pem | Used to specify the location of the file containing trusted CA certificates for authenticating the Hive server when using SSL.  
Note: This setting is only applicable to **User Name and Password** (SSL) and **HTTPS** authentication mechanisms, and will be ignored by other authentication mechanisms.  
Note: If this setting is not set, then the driver will default to using the trusted CA certificates file installed by the driver.  
(Optional) |
| ForceSynchronousExec   | 0                                                                 | Used to force the driver to do synchronous query execution when connected to a Hive cluster capable of asynchronous query execution.  
Set to 1 to enable.  
Set to 0 to disable.  
(Optional) |
### Appendix C: Driver Configuration Options

<table>
<thead>
<tr>
<th>Key</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DelegationUID</td>
<td></td>
<td>Used to delegate all operation against Hive to a user that is different than the authenticated user for the connection.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This setting is only applicable when connecting to a Hive Server 2 that supports this feature. Otherwise this setting will not take any effect.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Optional)</td>
</tr>
<tr>
<td>AsyncExecPollInterval</td>
<td>100</td>
<td>The time in millisecond between each poll for the status of an asynchronous query execution.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This configuration is only applicable to Hive 0.12.0 or above, and is ignored otherwise.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Asynchronous execution here doesn’t mean ODBC asynchronous operations are supported; it only means the RPC call used to execute a query against Hive is asynchronous.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Optional)</td>
</tr>
<tr>
<td>Key</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DriverConfigTakePrecedence</td>
<td>0</td>
<td>Allow driver wide configurations to take precedence over connection string and DSN settings. Set to 1 to enable. Set to 0 to disable. (Optional)</td>
</tr>
<tr>
<td>GetTablesWithQuery</td>
<td>0</td>
<td>Controls whether to retrieve the names of tables in a database using the GET TABLES query instead of the GetTables Thrift API call.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> This setting is only applicable when connecting to Hive Server 2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set to 1 to enable. Set to 0 to disable. (Optional)</td>
</tr>
<tr>
<td>EnableTempTable</td>
<td>0</td>
<td>Controls whether to enable or disable the temporary table feature. Set to 1 to enable. Set to 0 to disable. (Optional)</td>
</tr>
<tr>
<td>WebHDFSHost</td>
<td>The Hive Server host.</td>
<td>The hostname or IP address of the machine hosting both the NameNode of your Hadoop cluster and the WebHDFS service. (Optional)</td>
</tr>
<tr>
<td>WebHDFSPort</td>
<td>50070</td>
<td>The name of the HDFS user that the driver will use to create the necessary files for supporting the Temporary Table feature. (Optional)</td>
</tr>
</tbody>
</table>
### Appendix C: Driver Configuration Options

<table>
<thead>
<tr>
<th>Key</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDFSTempTableDir</td>
<td>/tmp/simba</td>
<td>The HDFS directory that the driver will use to store the necessary files for supporting the Temporary Table feature.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> Due to a problem in Hive (see <a href="https://issues.apache.org/jira/browse/HIVE-4554">https://issues.apache.org/jira/browse/HIVE-4554</a>), space characters in HDFS path will not work with versions of Hive prior to 0.12.0.</td>
</tr>
<tr>
<td>TempTableTTL</td>
<td>10</td>
<td>The number of minutes a temporary table is guaranteed to exist in Hive after it is created.</td>
</tr>
<tr>
<td>ShowHiveSystemTable</td>
<td>0</td>
<td>Controls whether the driver returns the HIVE_SYSTEM table for catalog function calls such as SQLTables and SQLColumns.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set to 0 to not return the HIVE_SYSTEM table.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set to 1 to return the HIVE_SYSTEM table.</td>
</tr>
<tr>
<td>UseUnicodeSqlCharacterTypes</td>
<td>0</td>
<td>Controls whether the driver returns SQL_WVARCHAR or SQL_VARCHAR for STRING and VARCHAR columns, and SQL_WCHAR or SQL_CHAR for CHAR columns.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set to 0 to return SQL_VARCHAR/SQL_CHAR.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set to 1 to return SQL_WVARCHAR/SQL_WCHAR.</td>
</tr>
</tbody>
</table>
Appendix D: Temporary Table CREATE TABLE and INSERT Statements

Temporary Table CREATE TABLE Statement

The following DDL syntax for creating temporary table is supported:

```
<create table statement> := CREATE TABLE <temporary table name> <left paren><column definition list><right paren>
<column definition list> := <column definition>[, <column definition>]*
<column definition> := <column name> <data type>
<temporary table name> := <double quote><number sign><table name><double quote>
<left paren> := ( 
<right paren> := )
<double quote> := " 
<number sign> := #
```

The following is an example of a CREATE TABLE SQL statement for creating a temporary table:

```
CREATE TABLE "#TEMPTABLE1" (C1 DATATYPE_1, C2 DATATYPE_2, ..., Cn DATATYPE_n)
```

**Note:**

The temporary table name in a SQL query must be surrounded by double quotes, and the name must begin with a number sign.

**Note:**

Data types are limited by the set of data types supported by Hive.

Temporary Table INSERT Statement

The following is the supported INSERT syntax for temporary table:

```
<insert statement> := INSERT INTO <temporary table name> <left paren><column name list><right paren> VALUES <left paren><literal value list><right paren>
<column name list> := <column name>[, <column name>]*
<literal value list> := <literal value>[, <literal value>]*
<temporary table name> := <double quote><number sign><table name><double quote>
```
The following is an example of temporary table INSERT statement:

```
INSERT INTO "#TEMPTABLE1" values (VAL(C1), VAL(C2) ... VAL(Cn) )
```

VAL(C1) is the literal value for the first column in the table, and VAL(Cn) is the literal value for the nth column in the table.

**Note:**

The INSERT statement is only supported for temporary tables.
### Conformance Level

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Core</td>
<td>SQLAllocHandle</td>
<td>Core</td>
<td>SQLGetStmtAttr</td>
</tr>
<tr>
<td>Core</td>
<td>SQLBindCol</td>
<td>Core</td>
<td>SQLGetTypeInfo</td>
</tr>
<tr>
<td>Core</td>
<td>SQLBindParameter</td>
<td>Core</td>
<td>SQLNativeSql</td>
</tr>
<tr>
<td>Core</td>
<td>SQLCancel</td>
<td>Core</td>
<td>SQLNumParams</td>
</tr>
<tr>
<td>Core</td>
<td>SQLCloseCursor</td>
<td>Core</td>
<td>SQLPrepare</td>
</tr>
<tr>
<td>Core</td>
<td>SQLColAttribute</td>
<td>Core</td>
<td>SQLParamData</td>
</tr>
<tr>
<td>Core</td>
<td>SQLColumns</td>
<td>Core</td>
<td>SQLPrepare</td>
</tr>
<tr>
<td>Core</td>
<td>SQLConnect</td>
<td>Core</td>
<td>SQLPutData</td>
</tr>
<tr>
<td>Core</td>
<td>SQLCopyDesc</td>
<td>Core</td>
<td>SQLRowCount</td>
</tr>
<tr>
<td>Core</td>
<td>SQLDescribeCol</td>
<td>Core</td>
<td>SQLSetConnectAttr</td>
</tr>
<tr>
<td>Core</td>
<td>SQLDisconnect</td>
<td>Core</td>
<td>SQLSetCursorName</td>
</tr>
<tr>
<td>Core</td>
<td>SQLEndTran</td>
<td>Core</td>
<td>SQLSetDescField</td>
</tr>
<tr>
<td>Core</td>
<td>SQLEXecDirect</td>
<td>Core</td>
<td>SQLSetEnvAttr</td>
</tr>
<tr>
<td>Core</td>
<td>SQLExecute</td>
<td>Core</td>
<td>SQLSetStmtAttr</td>
</tr>
<tr>
<td>Core</td>
<td>SQLFetch</td>
<td>Core</td>
<td>SQLSpecialColumns</td>
</tr>
<tr>
<td>Core</td>
<td>SQLFetchScroll</td>
<td>Core</td>
<td>SQLStatistics</td>
</tr>
<tr>
<td>Core</td>
<td>SQLFreeHandle</td>
<td>Core</td>
<td>SQLTables</td>
</tr>
<tr>
<td>Core</td>
<td>SQLFreeStmt</td>
<td>Core</td>
<td>SQLBrowseConnect</td>
</tr>
<tr>
<td>Core</td>
<td>SQLGetConnectAttr</td>
<td>Level 1</td>
<td>SQLPrimaryKeys</td>
</tr>
</tbody>
</table>

[^1]: ODBC Compliance levels are Core, Level 1 and Level 2. These are defined in the ODBC Specification published with the Interface SDK from Microsoft.

### Appendix E: ODBC API Conformance Level

<table>
<thead>
<tr>
<th>Conformance Level(^1)</th>
<th>INTERFACES(^2)</th>
<th>Conformance Level(^1)</th>
<th>INTERFACES(^2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core</td>
<td>SQLGetCursorName</td>
<td>Core</td>
<td>SQLGetInfo</td>
</tr>
<tr>
<td>Core</td>
<td>SQLGetData</td>
<td>Level 1</td>
<td>SQLProcedureColumns</td>
</tr>
<tr>
<td>Core</td>
<td>SQLGetDescField</td>
<td>Level 1</td>
<td>SQLProcedures</td>
</tr>
<tr>
<td>Core</td>
<td>SQLGetDescRec</td>
<td>Level 1</td>
<td>SQLProcedureColumns</td>
</tr>
<tr>
<td>Core</td>
<td>SQLGetDiagField</td>
<td>Level 2</td>
<td>SQLColumnPrivileges</td>
</tr>
<tr>
<td>Core</td>
<td>SQLGetDiagRec</td>
<td>Level 2</td>
<td>SQLDescribeParam</td>
</tr>
<tr>
<td>Core</td>
<td>SQLGetEnvAttr</td>
<td>Level 2</td>
<td>SQLForeignKeys</td>
</tr>
<tr>
<td>Core</td>
<td>SQLGetFunctions</td>
<td>Level 2</td>
<td>SQITablePrivileges</td>
</tr>
</tbody>
</table>

\(^1\) ODBC Compliance levels are Core, Level 1 and Level 2. These are defined in the ODBC Specification published by Microsoft.

\(^2\) Interfaces include both the Unicode and non-unicode versions. See [http://msdn.microsoft.com/en-us/library/ms716246%28VS.85%29.aspx](http://msdn.microsoft.com/en-us/library/ms716246%28VS.85%29.aspx) for more details.