

Cloudera ODBC Driver for Apache Hive Version 2.5.14



Important Notice

© 2010-2015 Cloudera, Inc. All rights reserved.

Cloudera, the Cloudera logo, Cloudera Impala, Impala, 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: 2.5.14

Date: February 13, 2015

Table of Contents

INTRODUCTION	1
WINDOWS DRIVER.....	1
SYSTEM REQUIREMENTS.....	1
INSTALLING THE DRIVER.....	2
VERIFYING THE VERSION NUMBER.....	2
CREATING A DATA SOURCE NAME (DSN)	3
CONFIGURING A DSN-LESS CONNECTION.....	5
CONFIGURING AUTHENTICATION.....	6
<i>Using No Authentication.....</i>	<i>7</i>
<i>Using Kerberos.....</i>	<i>7</i>
<i>Using User Name</i>	<i>8</i>
<i>Using User Name and Password.....</i>	<i>8</i>
<i>Using User Name and Password (SSL)</i>	<i>9</i>
<i>Using Windows Azure HDInsight Emulator.....</i>	<i>10</i>
<i>Using Windows Azure HDInsight Service</i>	<i>10</i>
<i>Using HTTP.....</i>	<i>10</i>
<i>Using HTTPS.....</i>	<i>10</i>
<i>Using Kerberos over HTTP.....</i>	<i>11</i>
<i>Using Kerberos over HTTPS.....</i>	<i>12</i>
CONFIGURING KERBEROS AUTHENTICATION FOR WINDOWS.....	13
<i>Active Directory.....</i>	<i>13</i>
<i>MIT Kerberos.....</i>	<i>13</i>
CONFIGURING ADVANCED OPTIONS.....	17
CONFIGURING SERVER-SIDE PROPERTIES	18
CONFIGURING THE TEMPORARY TABLE FEATURE	19
LINUX DRIVER.....	20
SYSTEM REQUIREMENTS.....	20
INSTALLING THE DRIVER.....	21
VERIFYING THE VERSION NUMBER.....	22
SETTING THE LD_LIBRARY_PATH ENVIRONMENT VARIABLE.....	22

MAC OS X DRIVER	23
SYSTEM REQUIREMENTS	23
INSTALLING THE DRIVER.....	23
VERIFYING THE VERSION NUMBER	24
SETTING THE DYLD_LIBRARY_PATH ENVIRONMENT VARIABLE	24
AIX DRIVER	24
SYSTEM REQUIREMENTS	24
INSTALLING THE DRIVER.....	25
VERIFYING THE VERSION NUMBER	25
SETTING THE LD_LIBRARY_PATH ENVIRONMENT VARIABLE.....	26
CONFIGURING ODBC CONNECTIONS FOR NON-WINDOWS PLATFORMS	26
FILES.....	26
SAMPLE FILES	26
CONFIGURING THE ENVIRONMENT.....	27
CONFIGURING THE ODBC.INI FILE.....	28
CONFIGURING THE ODBCINST.INI FILE.....	29
CONFIGURING THE CLOUDERA.HIVEODBC.INI FILE	30
CONFIGURING SERVICE DISCOVERY MODE	31
CONFIGURING AUTHENTICATION	31
<i>Using No Authentication.....</i>	<i>32</i>
<i>Using Kerberos</i>	<i>32</i>
<i>Using User Name</i>	<i>32</i>
<i>Using User Name and Password</i>	<i>33</i>
<i>Using User Name and Password (SSL)</i>	<i>33</i>
<i>Using HTTP.....</i>	<i>33</i>
<i>Using HTTPS.....</i>	<i>34</i>
<i>Using Kerberos over HTTP.....</i>	<i>35</i>
<i>Using Kerberos over HTTPS.....</i>	<i>35</i>
FEATURES	36
SQL QUERY VERSUS HIVEQL QUERY.....	36
SQL CONNECTOR.....	36
DATA TYPES	37

CATALOG AND SCHEMA SUPPORT.....	38
AUTHENTICATION.....	38
<i>Using No Authentication</i>	40
<i>Using Kerberos</i>	40
<i>Using User Name</i>	40
<i>Using User Name and Password</i>	40
<i>Using User Name and Password (SSL)</i>	40
<i>Using HTTP</i>	41
<i>Using HTTPS</i>	41
<i>Using Kerberos over HTTP</i>	41
<i>Using Kerberos over HTTPS</i>	41
HIVE_SYSTEM TABLE.....	41
SERVER-SIDE PROPERTIES.....	41
TEMPORARY TABLE.....	42
<i>CREATE TABLE Statement for Temporary Tables</i>	42
<i>INSERT Statement for Temporary Tables</i>	43
GET TABLES WITH QUERY.....	43
ACTIVE DIRECTORY.....	44
WRITE-BACK.....	44
DYNAMIC SERVICE DISCOVERY USING ZOOKEEPER.....	44
DRIVER CONFIGURATION OPTIONS.....	44
CONTACT US.....	56
APPENDIX A: USING A CONNECTION STRING.....	57
DSN CONNECTIONS.....	57
DSN-LESS CONNECTIONS.....	57
APPENDIX B: ODBC API CONFORMANCE LEVEL.....	58

Introduction

The 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, 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 36.

The 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.

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. For more information about ODBC, see <http://www.simba.com/odbc.htm>. For complete information about the ODBC specification, see the *ODBC API Reference* at [http://msdn.microsoft.com/en-us/library/windows/desktop/ms714562\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms714562(v=vs.85).aspx)

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 the 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 (Professional and Enterprise)
 - Windows® 8 (Pro and Enterprise)
 - 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.

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:

For an explanation of how to use ODBC on 64-bit editions of Windows, see

<http://www.simba.com/wp-content/uploads/2010/10/HOW-TO-32-bit-vs-64-bit-ODBC-Data-Source-Administrator.pdf>


To install the 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 **Change**, then browse to the desired folder, and then click **OK**. To accept the installation location, click **Next**
5. Click **Install**
6. When the installation completes, click **Finish**

Verifying the Version Number

If you need to verify the version of the Cloudera ODBC Driver for Hive that is installed on your Windows machine, you can find the version number in the Cloudera ODBC Driver for Hive DSN Setup dialog box.

To verify the version number:

1. Click the **Start** button , then click **All Programs**, and then click the **Cloudera ODBC Driver for Apache Hive 2.5** program group corresponding to the bitness of the client application accessing data in Hadoop / Hive, and then click **ODBC Administrator**
2. If you have already created a DSN for the driver, then select the DSN and click **Configure**

OR

If you have not created a DSN for the driver, then do the following:

- a. On the **User DSN** tab or the **System DSN** tab, click **Add**
- b. In the Create New Data Source dialog box, select **Cloudera ODBC Driver for Hive** and then click **Finish**


3. Locate the driver version number in the lower-left corner of the Cloudera ODBC Driver for Hive DSN Setup dialog box.

Creating a Data Source Name (DSN)

Typically, after installing the Cloudera ODBC Driver for Hive, you need to create a Data Source Name (DSN).

For information about DSN-less connections, see “Configuring a DSN-less Connection” on page 5.

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** program group corresponding to the bitness of the client application accessing data in Hadoop / Hive, and then click **ODBC Administrator**
2. In the ODBC Data Source Administrator, 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.
3. To create a DSN that only the user currently logged into Windows can use, click the **User DSN** tab.

OR

To create a DSN that all users who log into Windows can use, click the **System DSN** tab.

4. Click **Add**
5. In the Create New Data Source dialog box, select **Cloudera ODBC Driver for Apache Hive** and then click **Finish**
6. Use the options in the Cloudera ODBC Driver for Apache Hive DSN Setup dialog box to configure your DSN:
 - a. In the **Data Source Name** field, type a name for your DSN.
 - b. Optionally, in the **Description** field, type relevant details about the DSN.
 - c. To connect to Hive without using the Apache ZooKeeper service, in the **Service Discovery Mode** list, select **No Service Discovery**

OR

To enable the driver to discover Hive Server 2 services via the ZooKeeper service, in the **Service Discovery Mode** list, select **ZooKeeper**

- d. In the **Host(s)** field, if you selected **No Service Discovery** in step c, then type the IP address or host name of the Hive server.

OR

If you selected **ZooKeeper** in step c, then type a comma-separated list of ZooKeeper servers. Use the following format, where *zk_host* is the IP address or host name of the ZooKeeper server and *zk_port* is the number of the port that the ZooKeeper server uses:

```
zk_host1:zk_port1, zk_host2:zk_port2
```

- e. In the **Port** field, if you selected **No Service Discovery** in step c, then type the number of the TCP port on which the Hive server is listening. Otherwise, do not type a value in the field.
- f. In the **Database** field, type the name of the database schema to use when a schema is not explicitly specified in a query.

Note:

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

- g. In the **ZooKeeper Namespace** field, if you selected **ZooKeeper** in step c, then type the namespace on ZooKeeper under which Hive Server 2 znodes are added. Otherwise, do not type a value in the field.
- h. In the **Hive Server Type** list, select **Hive Server 1** or **Hive Server 2**

Note:

If you selected **ZooKeeper** in step c, then **Hive Server 1** is not supported.

- i. In the **Authentication** area, configure authentication as needed. For more information, see “Configuring Authentication” on page 6.

Note:

Hive Server 1 does not support authentication. Most default configurations of Hive Server 2 require **User Name** authentication. To verify the authentication mechanism that you need to use for your connection, check the configuration of your Hadoop / Hive distribution. For more information, see “Authentication” on page 38.

- j. Optionally, if the operations against Hive are to be done on behalf of a user that is different than the authenticated user for the connection, type the name of the user to be delegated in the **Delegation UID** field.

Note:

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

- k. To configure advanced driver options, click **Advanced Options**. For more information, see “Configuring Advanced Options” on page 17.
- l. To configure server-side properties, click **Advanced Options** and then click **Server Side Properties**. For more information, see “Configuring Server-Side Properties” on page 18.
- m. To configure the Temporary Table feature, click **Advanced Options** and then click **Temporary Table Configuration**. For more information, see “Temporary Table” on page 42 and “Configuring the Temporary Table Feature” on page 19.

Important:

When connecting to Hive 0.14 or later, the Temporary Tables feature is always enabled and you do not need to configure it in the driver.


7. To test the connection, click **Test**. Review the results as needed, and then click **OK**. If the connection fails, then confirm that the settings in the Cloudera ODBC Driver for Apache Hive DSN Setup dialog box are correct. Contact your Hive server administrator as needed.
8. To save your settings and close the Cloudera ODBC Driver for Apache Hive DSN Setup dialog box, click **OK**
9. To close the ODBC Data Source Administrator, click **OK**

For more information about the configuration options available in the Cloudera ODBC Driver for Hive, see “Driver Configuration Options” on page 44.

Configuring a DSN-less Connection

Some client applications provide support for connecting to a data source using a driver without a Data Source Name (DSN). To configure a DSN-less connection, you can use a connection string or the Cloudera Hive ODBC Driver Configuration tool that is installed with the Cloudera ODBC Driver for Hive. The following section explains how to use the driver configuration tool. For information about using connection strings, see “DSN-less Connections” on page 57.

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

1. Click the **Start** button , then click **All Programs**, then click the **Cloudera ODBC Driver for Apache Hive 2.5** program group corresponding to the bitness of the client application accessing data in Hadoop / Hive.
2. 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.

3. To connect to Hive without using the Apache ZooKeeper service, in the **Service Discovery Mode** list, select **No Service Discovery**

OR

To enable the driver to discover Hive Server 2 services via the ZooKeeper service, in the **Service Discovery Mode** list, select **ZooKeeper**

4. In the **ZooKeeper Namespace** field, if you selected **ZooKeeper** in step 3, then type the namespace on ZooKeeper under which Hive Server 2 znodes are added. Otherwise, do not type a value in the field.
5. In the **Hive Server Type** list, select **Hive Server 1** or **Hive Server 2**

Note:

If you selected **ZooKeeper** in step 3, then **Hive Server 1** is not supported.

6. In the **Authentication** area, configure authentication as needed. For more information, see “Configuring Authentication” on page 6.

Note:

Hive Server 1 does not support authentication. Most default configurations of Hive Server 2 require **User Name** authentication. To verify the authentication mechanism that you need to use for your connection, check the configuration of your Hadoop / Hive distribution. For more information, see “Authentication” on page 38.

7. Optionally, if the operations against Hive are to be done on behalf of a user that is different than the authenticated user for the connection, type the name of the user to be delegated in the **Delegation UID** field.

Note:

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

8. To configure advanced options, click **Advanced Options**. For more information, see “Configuring Advanced Options” on page 17.
9. To configure server-side properties, click **Advanced Options** and then click **Server Side Properties**. For more information, see “Configuring Server-Side Properties” on page 18.
10. To configure the Temporary Table feature, click **Advanced Options** and then click **Temporary Table Configuration**. For more information, see “Temporary Table” on page 42 and “Configuring the Temporary Table Feature” on page 19.

Important:

When connecting to Hive 0.14 or later, the Temporary Tables feature is always enabled and you do not need to configure it in the driver.

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

Configuring Authentication

ODBC applications that connect to Hive Server 2 using a DSN can pass in authentication credentials by defining them in the DSN. To configure authentication for a connection that uses a DSN, use the ODBC Data Source Administrator.

Normally, applications that are not Hive Server 2 aware and that connect using a DSN-less connection do not have a facility for passing authentication credentials to the Cloudera ODBC Driver for Hive for a

connection. However, the Cloudera Hive ODBC Driver Configuration tool enables you to configure authentication without using a DSN.

Important:

Credentials defined in a DSN take precedence over credentials configured using the driver configuration tool. Credentials configured using the driver configuration tool apply for all connections that are made using a DSN-less connection unless the client application is Hive Server 2 aware and requests credentials from the user.

For information about selecting the appropriate authentication mechanism to use, see “Authentication” on page 38.

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

OR

To access authentication options for a DSN-less connection, open the Cloudera Hive ODBC Driver Configuration tool.

2. In the **Mechanism** list, select **No Authentication**
3. To save your settings and close the dialog box, click **OK**

Using Kerberos

Kerberos must be installed and configured before you can use this authentication mechanism. For more information, see “Configuring Kerberos Authentication for Windows” on page 13.

Note:

This authentication mechanism is available only for Hive Server 2.

To configure Kerberos authentication:

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

OR

To access authentication options for a DSN-less connection, open the Cloudera Hive ODBC Driver Configuration tool.

2. In the **Mechanism** list, select **Kerberos**

Windows Driver

3. If your Kerberos setup does not define a default realm or if the realm of your Hive Server 2 host is not the default, then type the Kerberos realm of the Hive Server 2 host in the **Realm** field.

OR

To use the default realm defined in your Kerberos setup, leave the **Realm** field empty.

4. In the **Host FQDN** field, type the fully qualified domain name of the Hive Server 2 host.
5. In the **Service Name** field, type the service name of the Hive server.

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, contact your Hadoop administrator.

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

Using User Name

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

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

OR

To access authentication options for a DSN-less connection, open the Cloudera Hive ODBC Driver Configuration tool.

2. In the **Mechanism** list, select **User Name**
3. In the **User Name** field, type an appropriate user name for accessing the Hive server.
4. To save your settings and close the dialog box, click **OK**

Using User Name and Password

This authentication mechanism requires a user name and a password.

Note:

This authentication mechanism is available only for Hive Server 2.

To configure User Name and Password authentication:

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

OR

To access authentication options for a DSN-less connection, open the Cloudera Hive ODBC Driver Configuration tool.

2. In the **Mechanism** list, select **User Name and Password**
3. In the **User Name** field, type an appropriate user name for accessing the Hive server.
4. In the **Password** field, type the password corresponding to the user name you typed in step 3.
5. To save your settings and close the dialog box, click **OK**

Using User Name and Password (SSL)

This authentication mechanism uses SSL and requires a user name and a password. The driver accepts self-signed SSL certificates for this authentication mechanism.

Note:

This authentication mechanism is available only for Hive Server 2, and SSL support in Hive Server 2 is available only in Hive 0.13 and later.

To configure User Name and Password (SSL) authentication:

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

OR

To access authentication options for a DSN-less connection, open the Cloudera Hive ODBC Driver Configuration tool.

2. In the **Mechanism** list, select **User Name and Password (SSL)**
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.
5. Optionally, configure the driver to allow the common name of a CA-issued certificate to not match the host name of the Hive server by clicking **Advanced Options** and selecting the **Allow Common Name Host Name Mismatch** check box.

Note:

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

Windows Driver

6. To configure the driver to load SSL certificates from a specific file, click **Advanced Options** and type the path to the file in the **Trusted Certificates** field.

OR

To use the trusted CA certificates PEM file that is installed with the driver, leave the **Trusted Certificates** field empty.

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

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

This authentication mechanism enables you to connect to a Hive Server 2 running in HTTP mode.

Note:

This authentication mechanism is available only for Hive Server 2, and HTTP support in Hive Server 2 is available only in Hive 0.13 and later.

To configure HTTP authentication:

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

OR

To access authentication options for a DSN-less connection, open the Cloudera Hive ODBC Driver Configuration tool.

2. In the **Mechanism** list, select **HTTP**
3. In the **HTTP Path** field, type the partial URL corresponding to the Hive server.
4. In the **User Name** field, type an appropriate user name for accessing the Hive server.
5. In the **Password** field, type the password corresponding to the user name you typed in step 4.
6. To save your settings and close the dialog box, click **OK**

Using HTTPS

This authentication mechanism enables you to connect to a Hive Server 2 that is running in HTTP mode and has SSL enabled. The driver accepts self-signed SSL certificates for this authentication mechanism.

Note:

This authentication mechanism is available only for Hive Server 2, and HTTPS support in Hive Server 2 is available only in Hive 0.13 and later.

To configure HTTPS authentication:

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

OR

To access authentication options for a DSN-less connection, open the Cloudera Hive ODBC Driver Configuration tool.

2. In the **Mechanism** list, select **HTTPS**
3. In the **HTTP Path** field, type the partial URL corresponding to the Hive server.
4. In the **User Name** field, type an appropriate user name for accessing the Hive server.
5. In the **Password** field, type the password corresponding to the user name you typed in step 4.
6. Optionally, configure the driver to allow the common name of a CA-issued certificate to not match the host name of the Hive server by clicking **Advanced Options** and selecting the **Allow Common Name Host Name Mismatch** check box.

Note:

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

7. To configure the driver to load SSL certificates from a specific file, click **Advanced Options** and type the path to the file in the **Trusted Certificates** field.

OR

To use the trusted CA certificates PEM file that is installed with the driver, leave the **Trusted Certificates** field empty.

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

Using Kerberos over HTTP

Kerberos must be installed and configured before you can use this authentication mechanism. For more information, see “Configuring Kerberos Authentication for Windows” on page 13.

Note:

This authentication mechanism is available only for Hive Server 2, and Kerberos over HTTP support in Hive Server 2 is available only in Hive 0.13 and later.

To configure Kerberos over HTTP authentication:

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

OR

To access authentication options for a DSN-less connection, open the Cloudera Hive ODBC Driver Configuration tool.

Windows Driver

2. In the **Mechanism** list, select **Kerberos over HTTP**
3. If your Kerberos setup does not define a default realm or if the realm of your Hive Server 2 is not the default, then type the Kerberos realm of the Hive Server 2 host in the **Realm** field.

OR

To use the default realm defined in your Kerberos setup, leave the **Realm** field empty.

4. In the **Host FQDN** field, type the value for the fully qualified domain name of the Hive Server 2 host.
5. In the **Service Name** field, type the value for the service name of the Hive server.
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, contact your Hadoop administrator.
6. In the **HTTP Path** field, type the partial URL corresponding to the Hive server.
7. To save your settings and close the dialog box, click **OK**

Using Kerberos over HTTPS

Kerberos must be installed and configured before you can use this authentication mechanism. For more information, see “Configuring Kerberos Authentication for Windows” on page 13.

The driver accepts self-signed SSL certificates for this authentication mechanism.

Note:

This authentication mechanism is available only for Hive Server 2, and Kerberos over HTTPS support in Hive Server 2 is available only in Hive 0.13 and later.

To use Kerberos over HTTPS authentication:

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

OR

To access authentication options for a DSN-less connection, open the Cloudera Hive ODBC Driver Configuration tool.

2. In the **Mechanism** list, select **Kerberos over HTTPS**
3. If your Kerberos setup does not define a default realm or if the realm of your Hive Server 2 host is not the default, then type the Kerberos realm of the Hive Server 2 host in the **Realm** field.

OR

To use the default realm defined in your Kerberos setup, leave the **Realm** field empty.

4. In the **Host FQDN** field, type the value for the fully qualified domain name of the Hive Server 2 host.
5. In the **Service Name** field, type the value for the service name of the Hive server.

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, contact your Hadoop administrator.

6. In the **HTTP Path** field, type the partial URL corresponding to the Hive server.
7. Optionally, configure the driver to allow the common name of a CA-issued certificate to not match the host name of the Hive server by clicking **Advanced Options** and selecting the **Allow Common Name Host Name Mismatch** check box.

Note:

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

8. To configure the driver to load SSL certificates from a specific file, click **Advanced Options** and type the path to the file in the **Trusted Certificates** field.

OR

To use the trusted CA certificates PEM file that is installed with the driver, leave the **Trusted Certificates** field empty.

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

Configuring Kerberos Authentication for Windows

Active Directory

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

- MIT Kerberos is *not* installed on client Windows machine.
- 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

Downloading and installing MIT Kerberos for Windows 4.0.1

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

To download and install MIT Kerberos for Windows 4.0.1:

1. To download the Kerberos installer for 64-bit computers, 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>

The 64-bit installer includes both 32-bit and 64-bit libraries.

OR

Windows Driver

To download the Kerberos installer for 32-bit computers, use the following download link from the MIT Kerberos website: <http://web.mit.edu/kerberos/dist/kfw/4.0/kfw-4.0.1-i386.msi>

The 32-bit installer includes 32-bit libraries only.

2. To run the installer, double-click the .msi file that you downloaded in step 1.
3. Follow the instructions in the installer to complete the installation process.
4. When the installation completes, click **Finish**

Setting Up the Kerberos Configuration File

Settings for Kerberos are specified through a configuration file. You can set up the configuration file as a .ini file in the default location (the **C:\ProgramData\MIT\Kerberos5** directory) or as a .conf file in a custom location.

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

Note:

For more information on configuring Kerberos, refer to the MIT Kerberos documentation.

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

1. Obtain a **krb5.conf** configuration file from your Kerberos administrator.

OR

Obtain the configuration file from the following location on the machine that is hosting the Hive Server 2: **/etc/krb5.conf**


2. Rename the configuration file from **krb5.conf** to **krb5.ini**
3. Copy the **krb5.ini** file to the **C:\ProgramData\MIT\Kerberos5** directory and overwrite the empty sample file.

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

1. Obtain a **krb5.conf** configuration file from your Kerberos administrator.

OR

Obtain the configuration file from the following location on the machine that is hosting the Hive Server 2: **/etc/krb5.conf**


2. Place the **krb5.conf** file in an accessible directory and make note of the full path name.
3. Click the **Start** button , then right-click **Computer**, and then click **Properties**
4. Click **Advanced system settings**
5. In the System Properties dialog box, click the **Advanced** tab and then click **Environment Variables**
6. In the Environment Variables dialog box, under the **System variables** list, click **New**
7. In the New System Variable dialog box, in the **Variable name** field, type **KRB5_CONFIG**

8. In the **Variable value** field, type the absolute path to the **krb5.conf** file from step 2.
9. Click **OK** to save the new variable.
10. Ensure that the variable is listed in the **System variables** list.
11. Click **OK** to close the Environment Variables dialog box, and then click **OK** to close the System Properties dialog box.

Setting Up the Kerberos Credential Cache File

Kerberos uses a credential cache to store and manage credentials.

To set up the Kerberos credential cache file:

1. Create a directory where you want to save the Kerberos credential cache file.
For example, create the following directory: **C:\temp**
2. Click the **Start** button , then right-click **Computer**, and then click **Properties**
3. Click **Advanced system settings**
4. In the System Properties dialog box, click the **Advanced** tab and then click **Environment Variables**
5. In the Environment Variables dialog box, under the **System variables** list, click **New**
6. In the New System Variable dialog box, in the **Variable name** field, type **KRB5CCNAME**
7. In the **Variable value** field, type the path to the folder you created in step 1, and then append the file name **krb5cache**

For example, if you created the folder **C:\temp** in step 1, then type **C:\temp\krb5cache**

Note:


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

8. Click **OK** to save the new variable.
9. Ensure that the variable appears in the **System variables** list.
10. Click **OK** to close the Environment Variables dialog box, and then click **OK** to close the System Properties dialog box.
11. To ensure that Kerberos uses the new settings, restart your computer.

Obtaining a Ticket for a Kerberos Principal


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

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

1. Click the **Start** button , then click **All Programs**, and then click the **Kerberos for Windows (64-bit)** or the **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.

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

1. Click the **Start** button , then click **All Programs**, then click **Accessories**, and then click **Command Prompt**
2. In the Command Prompt, type a command using the following syntax:

```
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 user principal to use for authentication.

For example, hive/hiveserver2.example.com@EXAMPLE.COM

3. If the cache location KRB5CCNAME is not set or used, then use the **-c** option of the **kinit** command to specify the location of the credential cache.

In the command, the **-c** argument must appear last. For example:

```
kinit -k -t C:\mykeytabs\hive.keytab  
hive/hiveserver2.example.com@EXAMPLE.COM -c  
c:\ProgramData\MIT\krbcache
```

Note:

Krbcache is the Kerberos cache file, not a directory.

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

Note:

For information about configuring a default keytab file for your Kerberos configuration, refer to the MIT Kerberos documentation.

1. Click the **Start** button , then click **All Programs**, then click **Accessories**, and then click **Command Prompt**

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

```
kinit -k principal
```

principal is the Kerberos user principal to use for authentication.

For example, `hive/hiveserver2.example.com@EXAMPLE.COM`

- If the cache location `KRB5CCNAME` is not set or used, then use the `-c` option of the `kinit` command to specify the location of the credential cache.

In the command, the `-c` argument must appear last. For example:

```
kinit -k -t C:\mykeytabs\hive.keytab
hive/hiveserver2.example.com@EXAMPLE.COM -c
c:\ProgramData\MIT\krbcache
```

Note:

Krbcache is the Kerberos cache file, not a directory.

Configuring Advanced Options

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

To configure advanced options:

- To access advanced options for a DSN, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, and then click **Advanced Options**

OR

To access advanced options for a DSN-less connection, open the Cloudera Hive ODBC Driver Configuration tool, and then click **Advanced Options**

- To disable the SQL Connector feature, select the **Use Native Query** check box.
- To defer query execution to `SQLExecute`, select the **Fast SQLPrepare** check box.
- To allow driver-wide configurations to take precedence over connection and DSN settings, select the **Driver Config Take Precedence** check box.
- To use the asynchronous version of the API call against Hive for executing a query, select the **Use Async Exec** check box.

Note:

This option is applicable only when connecting to a Hive cluster running Hive 0.12.0 or later.

- To retrieve the names of tables in a database by using the `SHOW TABLES` query, select the **Get Tables With Query** check box.

Note:

This option is applicable only when connecting to Hive Server 2.

7. To enable the driver to return SQL_WVARCHAR instead of SQL_VARCHAR for STRING and VARCHAR columns, and SQL_WCHAR instead of SQL_CHAR for CHAR columns, select the **Unicode SQL character types** check box.
8. To enable the driver to return the HIVE_SYSTEM table for catalog function calls such as SQLTables and SQLColumns, select the **Show HIVE_SYSTEM Table** check box.
9. In the **Rows Fetched Per Block** field, type the number of rows to be fetched per block.
10. In the **Default String Column Length** field, type the maximum data length for STRING columns.
11. In the **Binary column length** field, type the maximum data length for BINARY columns.
12. In the **Decimal Column Scale** field, type the maximum number of digits to the right of the decimal point for numeric data types.
13. To allow the common name of a CA-issued SSL certificate to not match the host name of the Hive server, select the **Allow Common Name Hostname Mismatch** check box.

Note:

This setting is only applicable to the **User Name and Password (SSL) authentication** mechanism.

14. To configure the driver to load SSL certificates from a specific file, type the path to the file in the **Trusted Certificates** field.

OR

To use the trusted CA certificates PEM file that is installed with the driver, leave the **Trusted Certificates** field empty.

Note:

This is only applicable to the **User Name and Password (SSL), HTTPS, and Kerberos over HTTPS** authentication mechanisms.

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

Configuring Server-Side Properties

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

To configure server-side properties:

1. To access advanced options for a DSN, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, then click **Advanced Options**, and then click **Server Side Properties**

OR

To access advanced options for a DSN-less connection, open the Cloudera Hive ODBC Driver Configuration tool, then click **Advanced Options**, and then click **Server Side Properties**

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

Note:

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

3. To edit a server-side property, select the property from the list, then click **Edit**, then update the **Key** and **Value** fields as needed, and then click **OK**
4. To delete a server-side property, select the property from the list, and then click **Remove**. In the confirmation dialog box, click **Yes**
5. To configure the driver to apply each server-side property by executing a query when opening a session to the Hive server, select the **Apply Server Side Properties with Queries** check box.

OR

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

Note:

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

6. To force the driver to convert server-side property key names to all lower case characters, select the **Convert Key Name to Lower Case** check box.
7. To save your settings and close the Server Side Properties dialog box, click **OK**

Configuring the Temporary Table Feature

You can configure the driver to create temporary tables. For more information about this feature, see “Temporary Table” on page 42.

Important:

When connecting to Hive 0.14 or later, the Temporary Tables feature is always enabled and you do not need to configure it in the driver.

To configure the Temporary Table feature:

1. To access advanced options for a DSN, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, then click **Advanced Options**, and then click **Temporary Table Configuration**

OR

To access advanced options for a DSN-less connection, open the Cloudera Hive ODBC Driver Configuration tool, then click **Advanced Options**, and then click **Temporary Table Configuration**

2. To enable the Temporary Table feature, select the **Enable Temporary Table** check box.
3. In the **Web HDFS Host** field, type the host name or IP address of the machine hosting both the namenode of your Hadoop cluster and the WebHDFS service. If this field is left blank, then the host name of the Hive server will be used.
4. In the **Web HDFS Port** field, type the WebHDFS port for the namenode.
5. In the **HDFS User** field, type the name of the HDFS user that the driver will use to create the necessary files for supporting the Temporary Table feature.
6. In the **Data file HDFS dir** field, type the HDFS directory that the driver will use to store the necessary files for supporting the Temporary Table feature.

Note:

Due to a problem in Hive (see <https://issues.apache.org/jira/browse/HIVE-4554>), HDFS paths with space characters do not work with versions of Hive prior to 0.12.0.

7. In the **Temp Table TTL** field, type the number of minutes that a temporary table is guaranteed to exist in Hive after it is created.
8. To save your settings and close the Temporary Table Configuration dialog box, click **OK**

For information about the statement syntax used for temporary tables, see “CREATE TABLE Statement for Temporary Tables” on page 42 and “INSERT Statement for Temporary Tables” on page 43.

Linux Driver

System Requirements

You install the 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 distributions (32- and 64-bit editions are supported):
 - Red Hat® Enterprise Linux® (RHEL) 5.0 or 6.0
 - CentOS 5.0 or 6.0
 - SUSE Linux Enterprise Server (SLES) 11
- 45 MB of available disk space

- One of the following ODBC driver managers installed:
 - iODBC 3.52.7 or above
 - unixODBC 2.2.12 or above

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

Installing the Driver

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

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

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.

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

- **/opt/cloudera/hiveodbc/ErrorMessages** contains error message files required by the driver.
- **/opt/cloudera/hiveodbc/Setup** contains sample configuration files named `odbc.ini` and `odbcinst.ini`
- **/opt/cloudera/hiveodbc/lib/32** contains the 32-bit driver and the `cloudera.hiveodbc.ini` configuration file.
- **/opt/cloudera/hiveodbc/lib/64** contains the 64-bit driver and the `cloudera.hiveodbc.ini` configuration file.

To install the Cloudera ODBC Driver for Hive:

- In Red Hat Enterprise Linux or CentOS, 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:

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

OR

Linux Driver

In SUSE Linux Enterprise Server, 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:

```
zypper install RPMFileName
```

The 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.

Verifying the Version Number

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

To verify the version number:

- At the command prompt, run the following command:

```
yum list | grep ClouderaHiveODBC
```

OR

Run the following command:

```
rpm -qa | grep ClouderaHive ODBC
```

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

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:

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

For information about how to set environment variables permanently, refer to your Linux shell documentation.

For information about creating ODBC connections using the Cloudera ODBC Driver for Hive, see “Configuring ODBC Connections for Non-Windows Platforms” on page 26.

Mac OS X Driver

System Requirements

You install the 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:

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

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

Installing the Driver

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

- **/opt/cloudera/hiveodbc/ErrorMessages** contains error messages required by the driver.
- **/opt/cloudera/hiveodbc/Setup** contains sample configuration files named `odbc.ini` and `odbcinst.ini`
- **/opt/cloudera/hiveodbc/lib/universal** contains the driver and the `cloudera.hiveodbc.ini` configuration file.

To install the Cloudera ODBC Driver for Hive:

1. Double-click **ClouderaHiveODBC.dmg** to mount the disk image.
2. Double-click **ClouderaHiveODBC.pkg** to run the installer.
3. In the installer, click **Continue**
4. On the Software License Agreement screen, click **Continue**, and when the prompt appears, click **Agree** if you agree to the terms of the License Agreement.
5. Optionally, to change the installation location, click **Change Install Location**, select the desired location, and then click **Continue**
6. To accept the installation location and begin the installation, click **Install**
7. When the installation completes, click **Close**

AIX Driver

Verifying the Version Number

If you need to verify the version of the Cloudera ODBC Driver for Hive that is installed on your Mac OS X machine, you can query the version number through the Terminal.

To verify the version number:

- At the Terminal, run the following command, where *user* is your user name on the computer:

```
hiveodbc user$ pkgutil --info cloudera.hiveodbc
```

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

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
```

For information about how to set environment variables permanently, refer to your Mac OS X shell documentation.

For details on creating ODBC connections using Cloudera ODBC Driver for Hive, see “Configuring ODBC Connections for Non-Windows Platforms” on page 26.

AIX Driver

System Requirements

You install the 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:

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

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

Installing the Driver

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

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

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.

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

- **/opt/cloudera/hiveodbc/ErrorMessages** contains error message srequired by the driver.
- **/opt/cloudera/hiveodbc/Setup** contains sample configuration files named `odbc.ini` and `odbcinst.ini`
- **/opt/cloudera/hiveodbc/lib/32** contains the 32-bit driver and the `cloudera.hiveodbc.ini` configuration file.
- **/opt/cloudera/hiveodbc/lib/64** contains the 64-bit driver and the `cloudera.hiveodbc.ini` configuration file.

To install the Cloudera ODBC Driver for Hive:

- 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:

```
rpm --install RPMFileName
```

Verifying the Version Number

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

To verify the version number:

- At the command prompt, run the following command:

```
rpm -qa | grep ClouderaHiveODBC
```

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

Configuring ODBC Connections for Non-Windows Platforms

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:

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

For information about how to set environment variables permanently, refer to your Linux shell documentation.

For information about creating ODBC connections using the Cloudera ODBC Driver for Hive, see “Configuring ODBC Connections for Non-Windows Platforms” on page 26.

Configuring ODBC Connections for Non-Windows Platforms

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** is used to define ODBC data sources, and it is required for DSNs.
- **.odbcinst.ini** is used to define ODBC drivers, and it is optional.

Also, by default the Cloudera ODBC Driver for Hive is configured using the cloudera.hiveodbc.ini file, which is located in one of the following directories depending on the version of the driver that you are using:

- **/opt/cloudera/hiveodbc/lib/32** for the 32-bit driver on Linux or AIX
- **/opt/cloudera/hiveodbc/lib/64** for the 64-bit driver on Linux or AIX
- **/opt/cloudera/hiveodbc/lib/universal** for the driver on Mac OS X

The cloudera.hiveodbc.ini file is required.

You can set driver 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. For information about the configuration options available for controlling the behavior of DSNs that are using the Cloudera ODBC Driver for Hive, see “Driver Configuration Options” on page 44.

Sample Files

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

- odbc.ini
- odbcinst.ini

Configuring ODBC Connections for Non-Windows Platforms

These sample configuration files provide preset values for settings related to the Cloudera ODBC Driver for Hive.

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

Optionally, you can use three environment variables—`ODBCINI`, `ODBCSYSINI`, and `CLOUDERAHIVEINI`—to specify different locations for the `odbc.ini`, `odbcinst.ini`, and `cloudera.hiveodbc.ini` configuration files by doing the following:

- Set `ODBCINI` to point to your `odbc.ini` file.
- Set `ODBCSYSINI` to point to the directory containing the `odbcinst.ini` file.
- Set `CLOUDERAHIVEINI` to point to your `cloudera.hiveodbc.ini` file.

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:

```
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

Note:

If you are using a DSN-less connection, then you do not need to configure the `odbc.ini` file. For information about configuring a DSN-less connection, see “DSN-less Connections” on page 57.

ODBC Data Source Names (DSNs) 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.

The following is an example `odbc.ini` configuration file for Linux and AIX:

```
[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
```

MyHiveServer is the IP address or hostname of the Hive server.

The following 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
```

MyHiveServer is the IP address or hostname of the Hive server.

To create a Data Source Name (DSN):

1. Open the `.odbc.ini` configuration file in a text editor.
2. In the `[ODBC Data Sources]` section, add a new entry by typing the Data Source Name (DSN), then an equal sign (=), and then the driver name.

Configuring ODBC Connections for Non-Windows Platforms

3. In the .odbc.ini file, add a new section with a name that matches the DSN you specified in step 2, and then add configuration options to the section. Specify configuration options as key-value pairs.

Note:

Hive Server 1 does not support authentication. Most default configurations of Hive Server 2 require **User Name** authentication, which you configure by setting the AuthMech key to 2. To verify the authentication mechanism that you need to use for your connection, check the configuration of your Hadoop / Hive distribution. For more information, see “Authentication” on page 38.

4. Save the .odbc.ini configuration file.

For information about the configuration options available for controlling the behavior of DSNs that are using the Cloudera ODBC Driver for Hive, see “Driver Configuration Options” on page 44.

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 28.

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.

The following 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
```

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

```
[ODBC Drivers]
```

Configuring ODBC Connections for Non-Windows Platforms

```
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. In the [ODBC Drivers] section, add a new entry by typing the driver name and then typing **=Installed**

Note:

Type a symbolic name that you want to use to refer to the driver in connection strings or DSNs.

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 in the Setup directory. Specify configuration options as key-value pairs.
4. Save the .odbcinst.ini configuration file.

Configuring the cloudera.hiveodbc.ini File

The cloudera.hiveodbc.ini file contains configuration settings for the Cloudera ODBC Driver for Hive. Settings that you define in the cloudera.hiveodbc.ini file apply to all connections that use the driver.

To configure the 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 is usually **UTF-16** or **UTF-32**, depending on the ODBC driver manager you use. iODBC uses **UTF-32**, and unixODBC uses **UTF-16**. To determine the correct setting to use, refer to your ODBC Driver Manager documentation.

OR

If you are using AIX and the unixODBC driver manager, then set the value to **UTF-16**. If you are using AIX and the iODBC driver manager, then set the value to **UTF-16** for the 32-bit driver or **UTF-32** for 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. To determine the correct library to specify, refer to your ODBC driver manager documentation.

The configuration file defaults to the shared library for iODBC. In Linux and AIX, the shared library name for iODBC is libiodbcinst.so. In Mac OS X, the shared library name for iODBC is libiodbcinst.dylib.

Note:

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 and AIX, 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 Service Discovery Mode

You can configure the Cloudera ODBC Driver for Hive to discover Hive Server 2 services via ZooKeeper.

To enable service discovery via ZooKeeper:

1. Open the odbc.ini configuration file in a text editor.
2. Set the ServiceDiscoveryMode connection attribute to 1
3. Set the ZKNamespace connection attribute to specify the namespace on ZooKeeper under which Hive Server 2 znodes are added.
4. Set the Host connection attribute to specify the ZooKeeper ensemble as a comma-separated list of ZooKeeper servers. For example, type the following, where *zk_host* is the IP address or host name of the ZooKeeper server and *zk_port* is the number of the port that the ZooKeeper server uses:

```
zk_host1:zk_port1, zk_host2:zk_port2
```

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

Depending on whether service discovery mode is enabled or disabled, you may need to provide different connection attributes or values in your connection string or DSN. For more information about connection attributes, see “Driver Configuration Options” on page 44.

Configuring Authentication

You can select the type of authentication to use for a connection by defining the AuthMech connection attribute in a connection string or in a DSN (in the odbc.ini file). Depending on the authentication mechanism you use, there may be additional connection attributes that you must define. For more information about the attributes involved in configuring authentication, see “Driver Configuration Options” on page 44.

For information about selecting the appropriate authentication mechanism to use, see “Authentication” on page 38.

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:

- Set the AuthMech connection attribute to 0

Using Kerberos

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

Note:

This authentication mechanism is available only for Hive Server 2.

To configure Kerberos authentication:

1. Set the AuthMech connection attribute 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 attribute.

OR

To use the default realm defined in your Kerberos setup, do not set the KrbRealm attribute.

3. Set the KrbHostFQDN attribute to the fully qualified domain name of the Hive Server 2 host.
4. Set the KrbServiceName attribute 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 KrbServiceName should be set to **hive**. If you are unsure of the correct service name to use for your particular Hadoop deployment, contact your Hadoop administrator.

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

Using User Name and Password

This authentication mechanism requires a user name and a password.

Note:

This authentication mechanism is available only for Hive Server 2.

To configure User Name and Password authentication:

1. Set the AuthMech connection attribute to 3
2. Set the UID attribute to an appropriate user name for accessing the Hive server.
3. Set the PWD attribute to the password corresponding to the user name you provided in step 2.

Using User Name and Password (SSL)

This authentication mechanism uses SSL and requires a user name and a password. The driver accepts self-signed SSL certificates for this authentication mechanism.

Note:

This authentication mechanism is available only for Hive Server 2, and SSL support in Hive Server 2 is available only in Hive 0.13 and later.

To configure User Name and Password (SSL) authentication:

1. Set the AuthMech connection attribute to 4
2. Set the UID attribute to an appropriate user name for accessing the Hive server.
3. Set the PWD attribute to the password corresponding to the user name you provided in step 2.
4. Optionally, configure the driver to allow the common name of a CA-issued certificate to not match the host name of the Hive server by setting the CAIssuedCertNamesMismatch attribute to 1.

Note:

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

5. To configure the driver to load SSL certificates from a specific file, set the TrustedCerts attribute to the path of the file.

OR

To use the trusted CA certificates PEM file that is installed with the driver, do not specify a value for the TrustedCerts attribute.

Using HTTP

This authentication mechanism enables you to connect to a Hive Server 2 running in HTTP mode.

Note:

This authentication mechanism is available only for Hive Server 2, and HTTP support in Hive Server 2 is available only in Hive 0.13 and later.

To configure HTTP authentication:

1. Set the AuthMech connection attribute to 7
2. Set the HTTPPath attribute to the partial URL corresponding to the Hive server.
3. Set the UID attribute to an appropriate user name for accessing the Hive server.
4. Set the PWD attribute to the password corresponding to the user name you typed in step 3.

Using HTTPS

This authentication mechanism enables you to connect to a Hive Server 2 that is running in HTTP mode and has SSL enabled. The driver accepts self-signed SSL certificates for this authentication mechanism.

Note:

This authentication mechanism is available only for Hive Server 2, and HTTPS support in Hive Server 2 is available only in Hive 0.13 and later.

To configure HTTPS authentication:

1. Set the AuthMech connection attribute to 8
2. Set the HTTPPath attribute to the partial URL corresponding to the Hive server.
3. Set the UID attribute to an appropriate user name for accessing the Hive server.
4. Set the PWD attribute to the password corresponding to the user name you typed in step 3.
5. Optionally, configure the driver to allow the common name of a CA-issued certificate to not match the host name of the Hive server by setting the CAIssuedCertNamesMismatch attribute to 1.

Note:

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

6. To configure the driver to load SSL certificates from a specific file, set the TrustedCerts attribute to the path of the file.

OR

To use the trusted CA certificates PEM file that is installed with the driver, do not specify a value for the TrustedCerts attribute.

Using Kerberos over HTTP

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

Note:

This authentication mechanism is available only for Hive Server 2, and Kerberos over HTTP support in Hive Server 2 is available only in Hive 0.13 and later.

To configure Kerberos over HTTP authentication:

1. Set the AuthMech connection attribute 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 attribute.

OR

To use the default realm defined in your Kerberos setup, do not set the KrbRealm attribute.

3. Set the KrbHostFQDN attribute to the fully qualified domain name of the Hive Server 2 host.
4. Set the KrbServiceName attribute to the service name of the Hive server.

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

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

Using Kerberos over HTTPS

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

The driver accepts self-signed SSL certificates for this authentication mechanism.

Note:

This authentication mechanism is available only for Hive Server 2, and Kerberos over HTTPS support in Hive Server 2 is available only in Hive 0.13 and later.

To configure Kerberos over HTTPS authentication:

1. Set the AuthMech connection attribute 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 attribute.

OR

To use the default realm defined in your Kerberos setup, do not set the KrbRealm attribute.

3. Set the KrbHostFQDN attribute to the fully qualified domain name of the Hive Server 2 host.
4. Set the KrbServiceName attribute to the service name of the Hive server.

Features

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

5. Set the HTTPPath attribute to the partial URL corresponding to the Hive server.
6. Optionally, configure the driver to allow the common name of a CA-issued certificate to not match the host name of the Hive server by setting the CAIssuedCertNamesMismatch attribute to 1.

Note:

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

7. To configure the driver to load SSL certificates from a specific file, set the TrustedCerts attribute to the path of the file.

OR

To use the trusted CA certificates PEM file that is installed with the driver, do not specify a value for the TrustedCerts attribute.

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, the syntax is different enough that most applications do not work with native HiveQL.

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

Data Types

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

Table 1 lists the supported data type mappings.

Hive Type	SQL Type
TINYINT	SQL_TINYINT
SMALLINT	SQL_SMALLINT
INT	SQL_INTEGER
BIGINT	SQL_BIGINT
FLOAT	SQL_REAL
DOUBLE	SQL_DOUBLE
DECIMAL	SQL_DECIMAL
BOOLEAN	SQL_BIT
STRING	SQL_VARCHAR <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> <p>Note: SQL_WVARCHAR is returned instead if the Unicode SQL character types configuration option (the UseUnicodeSqlCharacterTypes key) is enabled.</p> </div>
TIMESTAMP	SQL_TYPE_TIMESTAMP
VARCHAR(n)	SQL_VARCHAR
DATE	SQL_TYPE_DATE
DECIMAL(p,s)	SQL_DECIMAL
CHAR(n)	SQL_CHAR <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> <p>Note: SQL_WCHAR is returned instead if the Unicode SQL character types configuration option (the UseUnicodeSqlCharacterTypes key) is enabled.</p> </div>
BINARY	SQL_VARBINARY

Table 1 Supported Data Types

Features

Note:

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

Catalog and Schema Support

The 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 schemas/databases, 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.

Authentication

Hive Server 1 does not support authentication. You must select **No Authentication** as the authentication mechanism.

Hive Server 2 supports the following authentication mechanisms:

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

Important:

The **Windows Azure HDInsight Emulator** and **Window Azure HDInsight Service** authentication mechanisms are not supported in **Cloudera’s Distribution Including Apache Hadoop**.

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`
- `hive.server2.enable.doAs`

Table 2 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 OR User Name and Password (SSL) Note: User Name and Password (SSL) can only be used if your Hive server is configured with SSL.

Table 2 Hive Authentication Mechanism Configurations

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.

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.

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

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.

Features

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.

For more information about authentication mechanisms, refer to the documentation for your Hadoop / Hive distribution. See also “Running Hadoop in Secure Mode” at 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 DSN or driver configuration must be an existing OS 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 DSN or driver configuration is ignored.

If the user name is not specified in the DSN or driver configuration, 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 that 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.

Using User Name and Password (SSL)

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

Using HTTP

When connecting to a Hive server of type Hive Server 2 that is configured to use the Thrift HTTP transport over a TCP socket, you must configure your connection to use **HTTP** authentication.

Using HTTPS

When connecting to a Hive server of type Hive Server 2 that is configured to use the Thrift HTTP transport over a SSL socket, you must configure your connection to use **HTTPS** authentication.

Using Kerberos over HTTP

When connecting to a Hive server of type Hive Server 2 that is configured to use Kerberos as the authentication mechanism and Thrift HTTP as the transport over a TCP socket, you must configure your connection to use **Kerberos over HTTP** authentication.

Using Kerberos over HTTPS

When connecting to a Hive server of type Hive Server 2 that is configured to use Kerberos as the authentication mechanism and Thrift HTTP as the transport over a SSL socket, you must configure your connection to use **Kerberos over HTTPS** authentication.

HIVE_SYSTEM Table

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

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

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

Server-Side Properties

The 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 that is established using the DSN.

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

For information about setting server-side properties when using the Windows driver, see “Configuring Server-Side Properties” on page 18. For information about setting server-side properties when using the driver on a non-Windows platform, see “Driver Configuration Options” on page 44.

Features

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.

CREATE TABLE Statement for Temporary Tables

The driver supports the following DDL syntax for creating temporary tables:

```
<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 SQL statement for creating a temporary table:

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

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

Note:

You can only use data types that are supported by Hive.

INSERT Statement for Temporary Tables

The driver supports the following DDL syntax for inserting data into temporary tables:

```

<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>

<left paren> := (
<right paren> := )
<double quote> := "
<number sign> := #

```

The following is an example of a SQL statement for inserting data into temporary tables:

```
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.

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.

Driver Configuration Options

Active Directory

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

- MIT Kerberos is *not* installed on client Windows machine.
- 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.

Write-back

The Cloudera ODBC Driver for Hive supports translation for INSERT, UPDATE, and DELETE syntax when connecting to a Hive Server 2 instance that is running Hive 0.14 or later.

Dynamic Service Discovery using ZooKeeper

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

For information about configuring this feature in the Windows driver, see “Creating a Data Source Name (DSN)” on page 3 or “Configuring a DSN-less Connection” on page 5. For information about configuring this feature when using the driver on a non-Windows platform, see “Configuring Service Discovery Mode” on page 31.

Driver Configuration Options

Table 3 lists the configuration options available in the Cloudera ODBC Driver for Hive alphabetically by field or button label. Options that do not appear in the user interface of the driver are listed alphabetically by key name at the end of the table.

When creating or configuring a connection from a Windows machine, the fields and buttons described in *Table 3* are available in the Cloudera Hive ODBC Driver Configuration tool and the following dialog boxes:

- The Cloudera ODBC Driver for Apache Hive DSN Setup dialog box
- The Advanced Options dialog box
- The Server Side Properties dialog box

When using a connection string or configuring a connection from a Linux, Mac OS X, or AIX machine, use the key names provided in *Table 3*.

Note:

You can pass in configuration options in your connection string or set them 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 passed in in the connection string or set in an `odbc.ini` file are specific to a connection. Configuration options passed in using the connection string take precedence over configuration options set in `odbc.ini`. Configuration options set in `odbc.ini` take precedence over configuration options set in `cloudera.hiveodbc.ini`

Field or Button Label (Key Name)	Default Value	Description
Allow Common Name Host Name Mismatch (CAIssuedCertNamesMismatch)	Clear (0)	<p>When this option is enabled (1), the driver allows a CA-issued SSL certificate name to not match the host name of the Hive server.</p> <p>When this option is disabled (0), the CA-issued SSL certificate name must match the host name of the Hive server.</p> <div data-bbox="915 1018 1432 1272" style="border: 1px solid black; padding: 5px;"> <p>Note:</p> <p>This option is applicable only to the User Name and Password (SSL), HTTPS, and Kerberos over HTTPS authentication mechanisms.</p> </div> <p>(Optional)</p>

Driver Configuration Options

Field or Button Label (Key Name)	Default Value	Description
Apply properties with queries (ApplySSPWithQueries)	Selected (1)	<p>When this option is enabled (1), the driver applies each server-side property by executing a set SSPKey=SSPValue query when opening a session to the Hive server. When this option is disabled (0), the driver uses a more efficient method for applying server-side properties that does not involve additional network round-tripping. However, some Hive Server 2 builds are not compatible with the more efficient method.</p> <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> <p>Note: When connecting to a Hive Server 1, ApplySSPWithQueries is always enabled.</p> </div> <p>(Optional)</p>
Binary column length (BinaryColumnLength)	32767	<p>The maximum data length for BINARY columns.</p> <p>By default, the columns metadata for Hive does not specify a maximum data length for BINARY columns.</p> <p>(Optional)</p>
Convert Key Name to Lower Case (LCaseSspKeyName)	Selected (1)	<p>When this option is enabled (1), the driver converts server-side property key names to all lower case characters.</p> <p>When this option is disabled (0), the driver does not modify the server-side property key names.</p> <p>(Optional)</p>

Field or Button Label (Key Name)	Default Value	Description
Data file HFDS dir (HDFSTempTableDir)	/tmp/simba	<p>The HDFS directory that the driver will use to store the necessary files for supporting the Temporary Table feature.</p> <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> <p>Note:</p> <p>Due to a problem in Hive (see https://issues.apache.org/jira/browse/HIVE-4554), HDFS paths with space characters do not work with versions of Hive prior to 0.12.0.</p> </div> <p>(Optional) (Not applicable when connecting to Hive 0.14 or later)</p>
Database (Schema)	default	<p>The name of the database schema to use when a schema is not explicitly specified in a query. You can still issue queries on other schemas by explicitly specifying the schema in the query.</p> <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> <p>Note:</p> <p>To inspect your databases and determine the appropriate schema to use, type the show databases command at the Hive command prompt.</p> </div> <p>(Optional)</p>
Decimal column scale (DecimalColumnScale)	10	<p>The maximum number of digits to the right of the decimal point for numeric data types.</p> <p>(Optional)</p>
Default string column length (DefaultStringColumnLength)	255	<p>The maximum data length for STRING columns.</p> <p>By default, the columns metadata for Hive does not specify a maximum data length for STRING columns.</p> <p>(Optional)</p>

Driver Configuration Options

Field or Button Label (Key Name)	Default Value	Description
Delegation UID (DelegationUID)	None	<p>Use this option to delegate all operations against Hive to a user that is different than the authenticated user for the connection.</p> <div style="border: 1px solid orange; padding: 5px;"> <p>Note: This option is applicable only when connecting to a Hive Server 2 that supports this feature.</p> </div> <p>(Optional)</p>
Driver Config Take Precedence (DriverConfigTakePrecedence)	Clear (0)	<p>When this option is enabled (1), driver-wide configurations take precedence over connection and DSN settings.</p> <p>When this option is disabled (0), connection and DSN settings take precedence instead.</p> <p>(Optional)</p>
Enable Temporary Table (EnableTempTable)	Clear (0)	<p>When this option is enabled (1), the driver supports the creation and use of temporary tables.</p> <p>When this option is disabled (0), the driver does not support temporary tables.</p> <div style="border: 1px solid orange; padding: 5px;"> <p>Important: When connecting to Hive 0.14 or later, the Temporary Tables feature is always enabled and you do not need to configure it in the driver.</p> </div> <p>(Optional)</p> <p>(Not applicable when connecting to Hive 0.14 or later)</p>

Field or Button Label (Key Name)	Default Value	Description
Fast SQLPrepare (FastSQLPrepare)	Clear (0)	<p>When this option is enabled (1), the driver defers query execution to SQLExecute.</p> <p>When this option is disabled (0), the driver does not defer query execution to SQLExecute.</p> <p>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 Fast SQLPrepare.</p> <p>(Optional)</p>
Get Tables With Query (GetTablesWithQuery)	Clear (0)	<p>When this option is enabled (1), the driver uses the SHOW TABLES query to retrieve the names of the tables in a database.</p> <p>When this option is disabled (0), the driver uses the GetTables Thrift API call to retrieve the names of the tables in a database.</p> <div data-bbox="915 1066 1432 1199" style="border: 1px solid orange; padding: 5px;"> <p>Note: This option is applicable only when connecting to Hive Server 2.</p> </div> <p>(Optional)</p>
HDFS User (HDFSUser)	hdfs	<p>The name of the HDFS user that the driver will use to create the necessary files for supporting the Temporary Tables feature.</p> <p>(Optional)</p> <p>(Not applicable when connecting to Hive 0.14 or later)</p>

Driver Configuration Options

Field or Button Label (Key Name)	Default Value	Description
Hive Server Type (HiveServerType)	Hive Server 2 (2)	<p>The Hive server type.</p> <p>Select Hive Server 1 or set the key to 1 if you are connecting to a Hive Server 1 instance.</p> <p>Select Hive Server 2 or set the key to 2 if you are connecting to a Hive Server 2 instance.</p> <div style="border: 1px solid orange; padding: 5px; margin: 10px 0;"> <p>Note: If ZooKeeper is selected for Service Discovery Mode, then connections to Hive Server 1 are not supported.</p> </div> <p>(Optional)</p>
Host(s) (HOST)	None	<p>If Service Discovery Mode is set to No Service Discovery (0), specify the IP address or host name of the Hive server.</p> <p>If Service Discovery Mode is set to ZooKeeper (1), specify a comma-separated list of ZooKeeper servers in the following format, where <i>zk_host</i> is the IP address or host name of the ZooKeeper server and <i>zk_port</i> is the number of the port that the ZooKeeper server uses:</p> <div style="border: 1px dashed black; padding: 5px; margin: 10px 0;"> <pre>zk_host1:zk_port1, zk_host2: zk_port2</pre> </div> <p>(Required)</p>
Host FQDN (KrbHostFQDN)	None	<p>The fully qualified domain name of the Hive Server 2 host.</p> <p>(Required if the authentication mechanism is Kerberos, Kerberos over HTTP, or Kerberos over HTTPS)</p>
HTTP Path (HTTPPath)	None	<p>The partial URL corresponding to the Hive server on HTTP or HTTPS authentication mechanisms.</p> <p>(Required if the authentication mechanism is HTTP or HTTPS)</p>

Field or Button Label (Key Name)	Default Value	Description
Mechanism (AuthMech)	User Name (2)	<p>The authentication mechanism to use. Select one of the following settings, or set the key to the corresponding number:</p> <ul style="list-style-type: none"> • No Authentication (0) • Kerberos (1) • User Name (2) • User Name and Password (3) • User Name and Password (SSL) (4) • HTTP (7) • HTTPS (8) • Kerberos over HTTP (9) • Kerberos over HTTPS (10) <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> <p>Note: Do not set the key to 5 or 6; those authentication mechanisms are not supported in Cloudera's Distribution Including Apache Hadoop.</p> </div> <p>(Optional)</p>
Password (PWD)	None	<p>The password corresponding to the user name that you provided in the User Name field (the UID key).</p> <p>(Required if the authentication mechanism is User Name and Password, User Name and Password (SSL), HTTP, or HTTPS)</p>
Port (PORT)	10000	<p>The number of the TCP port on which the Hive server is listening.</p> <p>(Required if Service Discovery Mode is set to No Service Discovery (0))</p>
Realm (KrbRealm)	Depends on your Kerberos configuration.	<p>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.</p> <p>(Optional)</p>

Driver Configuration Options

Field or Button Label (Key Name)	Default Value	Description
Rows fetched per block (RowsFetchedPerBlock)	10000	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)
Service Discovery Mode (ServiceDiscoveryMode)	No Service Discovery (0)	When this option is enabled (1), the driver discovers Hive Server 2 services via the ZooKeeper service. When this option is disabled (0), the driver connects to Hive without using the ZooKeeper service. (Optional)
Service Name (KrbServiceName)	None	The Kerberos service principal name of the Hive server. (Required if the authentication mechanism is Kerberos , Kerberos over HTTP , or Kerberos over HTTPS)
Show HIVE_SYSTEM Table (ShowHiveSystemTable)	Clear (0)	When this option is enabled (1), the driver returns the HIVE_SYSTEM table for catalog function calls such as SQLTables and SQLColumns . When this option is disabled (0), the driver does not return the HIVE_SYSTEM table for catalog function calls. (Optional)
Temp Table TTL (TempTableTTL)	10	The number of minute a temporary table is guaranteed to exist in Hive after it is created. (Optional) (Not applicable when connecting to Hive 0.14 or later)

Field or Button Label (Key Name)	Default Value	Description
Trusted Certificates (TrustedCerts)	<p>The cacerts.pem file in the lib folder or subfolder within the driver’s installation directory.</p> <p>The exact file path varies depending on the version of the driver that is installed. For example, the path for the Windows driver is different from the path for the Mac OS X driver.</p>	<p>The location of the PEM file containing trusted CA certificates for authenticating the Hive server when using SSL.</p> <p>If this option is not set, then the driver will default to using the trusted CA certificates PEM file installed by the driver.</p> <div data-bbox="919 527 1430 751" style="border: 1px solid black; padding: 5px;"> <p>Note: This option is applicable only to User Name and Password (SSL), HTTPS, and Kerberos over HTTPS authentication mechanisms.</p> </div> <p>(Optional)</p>
Unicode SQL character types (UseUnicodeSqlCharacterTypes)	Clear (0)	<p>When this option is enabled (1), the driver returns SQL_WVARCHAR for STRING and VARCHAR columns, and returns SQL_WCHAR for CHAR columns.</p> <p>When this option is disabled (0), the driver returns SQL_VARCHAR for STRING and VARCHAR columns, and returns SQL_CHAR for CHAR columns.</p> <p>(Optional)</p>
Use Async Exec (EnableAsyncExec)	Clear (0)	<p>When this option is enabled (1), the driver uses an asynchronous version of the API call against Hive for executing a query.</p> <p>When this option is disabled (0), the driver executes queries synchronously.</p> <p>Due to a problem in Hive 0.12.0 (see 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.</p> <div data-bbox="919 1661 1430 1850" style="border: 1px solid black; padding: 5px;"> <p>Note: This option only takes effect when connecting to a Hive cluster running Hive 0.12.0 or higher.</p> </div> <p>(Optional)</p>

Driver Configuration Options

Field or Button Label (Key Name)	Default Value	Description
Use Native Query (UseNativeQuery)	Clear (0)	<p>When this option is enabled (1), the driver does not transform the queries emitted by an application, so the native query is used.</p> <p>When this option is disabled (0), the driver transforms the queries emitted by an application and converts them into an equivalent form in HiveQL.</p> <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> <p>Note: If the application is Hive-aware and already emits HiveQL, then enable this option to avoid the extra overhead of query transformation.</p> </div> <p>(Optional)</p>
User Name (UID)	anonymous (for User Name authentication only)	<p>The user name that you use to access Hive Server 2.</p> <p>(Optional if the authentication mechanism is User Name)</p> <p>(Required if the authentication mechanism is User Name and Password, User Name and Password (SSL), HTTP, or HTTPS)</p>
Web HDFS Host (WebHDFSHost)	The Hive server host.	<p>The host name or IP address of the machine hosting both the namenode of your Hadoop cluster and the WebHDFS service.</p> <p>(Optional)</p> <p>(Not applicable when connecting to Hive 0.14 or later)</p>
Web HDFS Port (WebHDFSPort)	50070	<p>The WebHDFS port for the namenode.</p> <p>(Optional)</p> <p>(Not applicable when connecting to Hive 0.14 or later)</p>
ZooKeeper Namespace (ZKNamespace)	None	<p>The namespace on ZooKeeper under which Hive Server 2 znodes are added.</p> <p>(Required if Service Discovery Mode is set to ZooKeeper (1))</p>

Field or Button Label (Key Name)	Default Value	Description
N/A (Driver)	The default value varies depending on the version of the driver that is installed. For example, the value for the Windows driver is different from the value for the Mac OS X driver.	The name of the installed driver (Cloudera Hive ODBC Driver) or the absolute path of the Cloudera ODBC Driver for Hive shared object file. (Required)
N/A (SSP_)	None	<p>Set a server-side property by using the following syntax, where <i>SSPKey</i> is the name of the server-side property to set and <i>SSPValue</i> is the value to assign to the server-side property:</p> <p>SSP_<i>SSPKey</i>=<i>SSPValue</i></p> <p>For example: SSP_mapred.queue.names=myQueue</p> <p>After the driver applies the server-side property, the SSP_ prefix is removed from the DSN entry, leaving an entry of SSPKey=SSPValue</p> <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> <p>Important: The SSP_ prefix must be upper case.</p> </div> <p>(Optional)</p>

Table 3 Driver Configuration Options

Contact Us

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: Using a Connection String

For some applications, you may need to use a connection string to connect to your data source.

DSN Connections

The following is an example of a connection string for a connection that uses a DSN:

```
DSN=DataSourceName; Key=Value
```

DataSourceName is the DSN that you are using for the connection. *Key* is any connection attribute that is not already specified as a configuration key in the DSN, and *Value* is the value for the attribute. Add key-value pairs to the connection string as needed, separating each pair with a semicolon (;). For information about the connection attributes that are available, see “Driver Configuration Options” on page 44.

DSN-less Connections

Some client applications provide support for connecting to a data source using a driver without a DSN. The following is an example of a connection string for a DSN-less connection:

```
Driver=DriverNameOrFile;HOST=MyHiveServer;PORT=PortNumber;  
Schema=DefaultSchema;HiveServerType=ServerType
```

The placeholders in the connection string are defined as follows:

- *DriverNameOrFile* is either the symbolic name of the installed driver defined in the `odbcinst.ini` file or the absolute path of the shared object file for the driver. If you use the symbolic name, then you must ensure that the `odbcinst.ini` file is configured to point the symbolic name to the shared object file. For more information, see “Configuring the `odbcinst.ini` File” on page 29.
- *MyHiveServer* is the IP address or host name of the Hive Server.
- *PortNumber* is the number of the port that the Hive server uses.
- *DefaultSchema* is the database schema to use when a schema is not explicitly specified in a query.
- *ServerType* is either 1 (for Hive Server 1) or 2 (for Hive Server 2).

Add key-value pairs to the connection string as needed, separating each pair with a semicolon (;). For information about the connection attributes that are available, see “Driver Configuration Options” on page 44.

Appendix B: ODBC API Conformance Level

Conformance Level ^[1]	INTERFACES ^[2]		Conformance Level ¹	INTERFACES ^[2]
Core	SQLAllocHandle		Core	SQLGetStmtAttr
Core	SQLBindCol		Core	SQLGetTypeInfo
Core	SQLBindParameter		Core	SQLNativeSql
Core	SQLCancel		Core	SQLNumParams
Core	SQLCloseCursor		Core	SQLNumResultCols
Core	SQLColAttribute		Core	SQLParamData
Core	SQLColumns		Core	SQLPrepare
Core	SQLConnect		Core	SQLPutData
Core	SQLCopyDesc		Core	SQLRowCount
Core	SQLDescribeCol		Core	SQLSetConnectAttr
Core	SQLDisconnect		Core	SQLSetCursorName
Core	SQLDriverconnect		Core	SQLSetDescField
Core	SQLEndTran		Core	SQLSetDescRec
Core	SQLExecDirect		Core	SQLSetEnvAttr
Core	SQLExecute		Core	SQLSetStmtAttr
Core	SQLFetch		Core	SQLSpecialColumns
Core	SQLFetchScroll		Core	SQLStatistics
Core	SQLFreeHandle		Core	SQLTables
Core	SQLFreeStmt		Core	SQLBrowseConnect
Core	SQLGetConnectAttr		Level 1	SQLPrimaryKeys

^[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.

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

Appendix B: ODBC API Conformance Level

Conformance Level ^[1]	INTERFACES ^[2]		Conformance Level ¹	INTERFACES ^[2]
Core	SQLGetCursorName		Core	SQLGetInfo
Core	SQLGetData		Level 1	SQLProcedureColumns
Core	SQLGetDescField		Level 1	SQLProcedures
Core	SQLGetDescRec		Level 1	SQLProcedureColumns
Core	SQLGetDiagField		Level 2	SQLColumnPrivileges
Core	SQLGetDiagRec		Level 2	SQLDescribeParam
Core	SQLGetEnvAttr		Level 2	SQLForeignKeys
Core	SQLGetFunctions		Level 2	SQLTablePrivileges

^[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> for more details.