Hortonworks Architecting the Future of Big Data

Hive ODBC Driver

User Guide

Revised: July 22, 2014

Table of Contents

Introduction

Welcome to the Hortonworks Hive ODBC Driver with SQL Connector. 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.

The Hortonworks Hive ODBC Driver with SQL Connector is used for direct SQL and HiveQL access to Apache Hadoop / Hive distributions. It enables Business Intelligence (BI), analytics and reporting on Hadoop / Hive-based data. The Hortonworks Hive ODBC Driver efficiently transforms an application's SQL query into the equivalent form in HiveQL. The Hive Query Language is a subset of SQL-92. If an application is Hive-aware, the Hortonworks Hive ODBC Driver is configurable to pass the query through. The Hortonworks Hive ODBC Driver with SQL Connector interrogates Hive to obtain schema information to present to a SQL-based application. Queries, including joins, are translated from SQL to HiveQL. For more information about the differences between HiveQL and SQL, refer to the Features section of this document.

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

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

Contact Us

If you have difficulty using the Hortonworks Hive ODBC Driver with SQL Connector, please contact our support staff. We welcome your questions, comments, and feature requests.

Please have a detailed summary of the client and server environment (OS version, patch-level, Hadoop distribution version, Hive version, configuration etc.) ready, before you call or write us. Supplying this information accelerates support.

By telephone:

USA: (855) 8-HORTON
International: (408) 916-4121

On the Internet:

Visit us at www.hortonworks.com.

Windows Driver

System Requirements

- Windows® XP with SP3, Windows® Vista, Windows® 7 Professional or Windows® 2008 R2. Both 32-bit and 64-bit editions are supported.
- 25 MB of available disk space.

Installing the driver requires administrator privileges.

The Hortonworks Hive ODBC Driver with SQL Connector requires a Hadoop cluster with the Hive service installed and running. The Hortonworks Hive ODBC Driver with SQL Connector is suitable for use with all versions of Apache Hive.

Installation

There are two versions of the driver for Windows:

- **HortonworksHiveODBC32.msi** for 32-bit
- **HortonworksHiveODBC64.msi** for 64-bit

The version of the driver that you select should match the bitness of the application. For example, if the application is 64-bit then you should install the 64-bit driver. It is allowable to install both versions of the driver.

The following document explains how to use ODBC on 64-bit editions of Windows: http://www.simba.com/wp-content/uploads/2010/10/HOW-TO-32-bit-vs-64-bit-ODBC-Data-Source-Administrator.pdf.

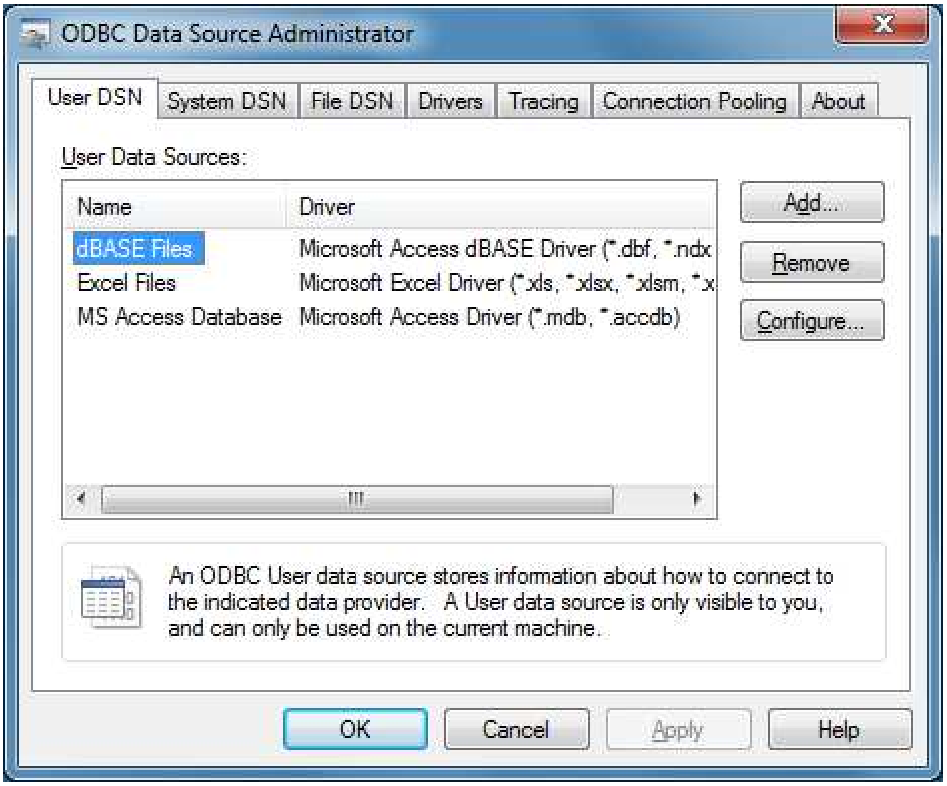
Configuration

Create a Data Source Name (DSN)

1. Click the **Start** button .
2. Click **All Programs.**
3. Click the **Hortonworks Hive ODBC Driver 1.4 (64-bit)** or the **Hortonworks Hive ODBC Driver 1.4 (32-bit)** program group. If you installed both versions of the driver, you will see two program groups.

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

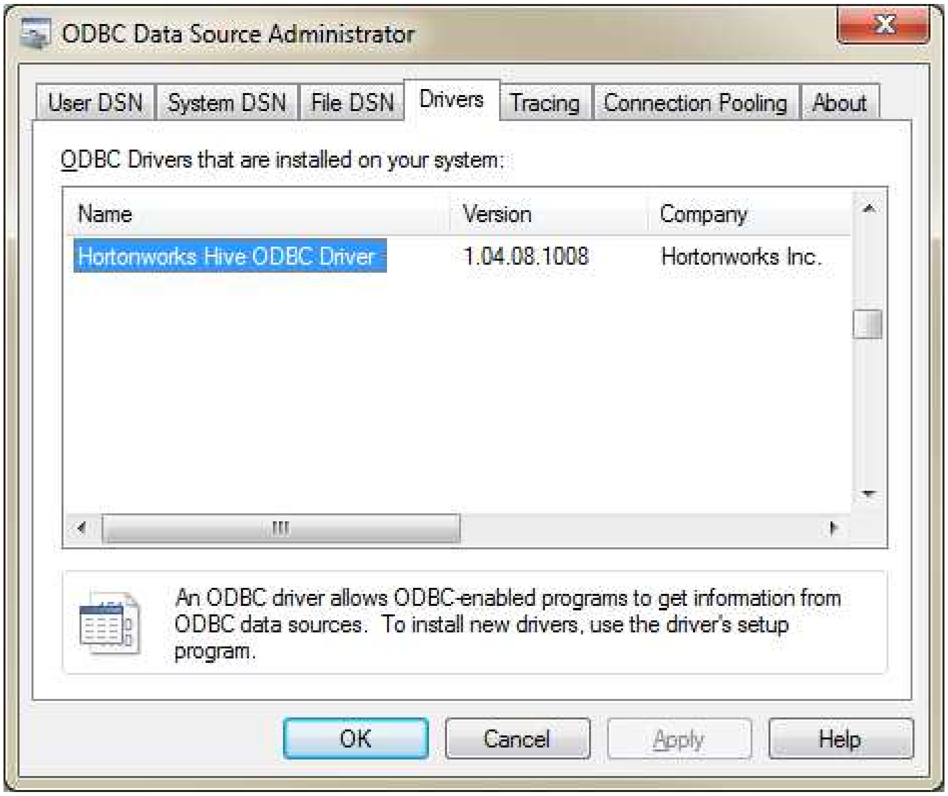
4. Click **64-bit ODBC Administrator or 32-bit ODBC Administrator**.
   The ODBC Data Source Administrator window opens.



Name	Driver
dBASE Files	Microsoft Access dBASE Driver (*.dbf, *.ndx
Excel Files	Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.x
MS Access Database	Microsoft Access Driver (*.mdb, *.accdb)

An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you, and can only be used on the current machine.

5. Click the **Drivers** tab and verify that the Hortonworks Hive ODBC Driver is displayed in the list of ODBC drivers that are installed on your system.



ODBC Drivers that are installed on your system:

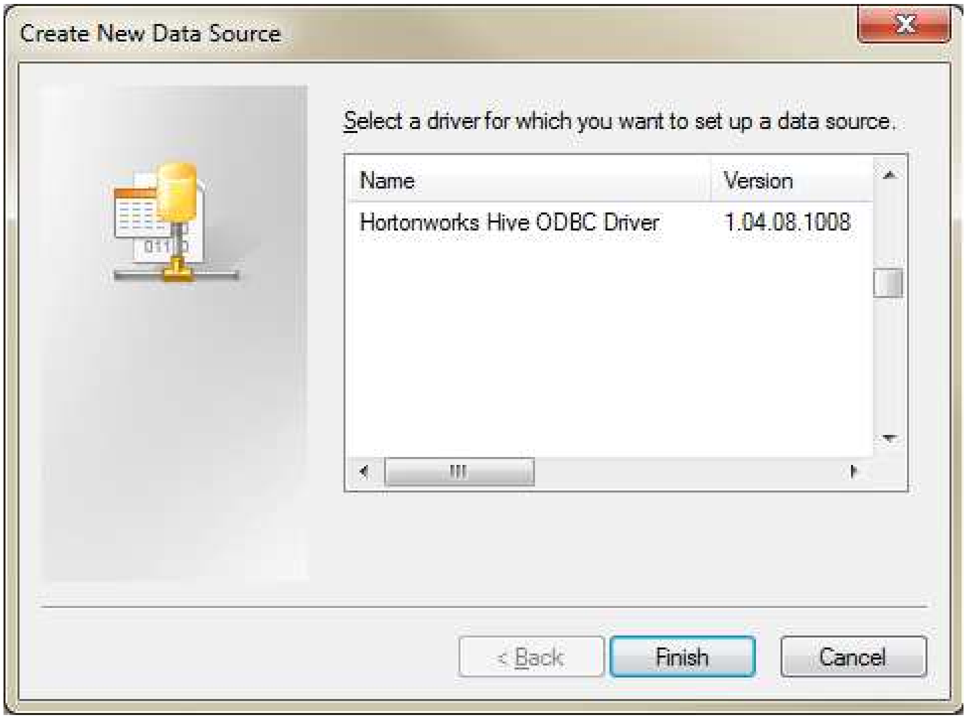
Name	Version	Company
Hortonworks Hive ODBC Driver	1.04.08.1008	Hortonworks Inc.

An ODBC driver allows ODBC-enabled programs to get information from ODBC data sources. To install new drivers, use the driver's setup program.

6. Click the **System DSN** tab to create a system DSN or click the **User DSN** tab to create a user DSN.

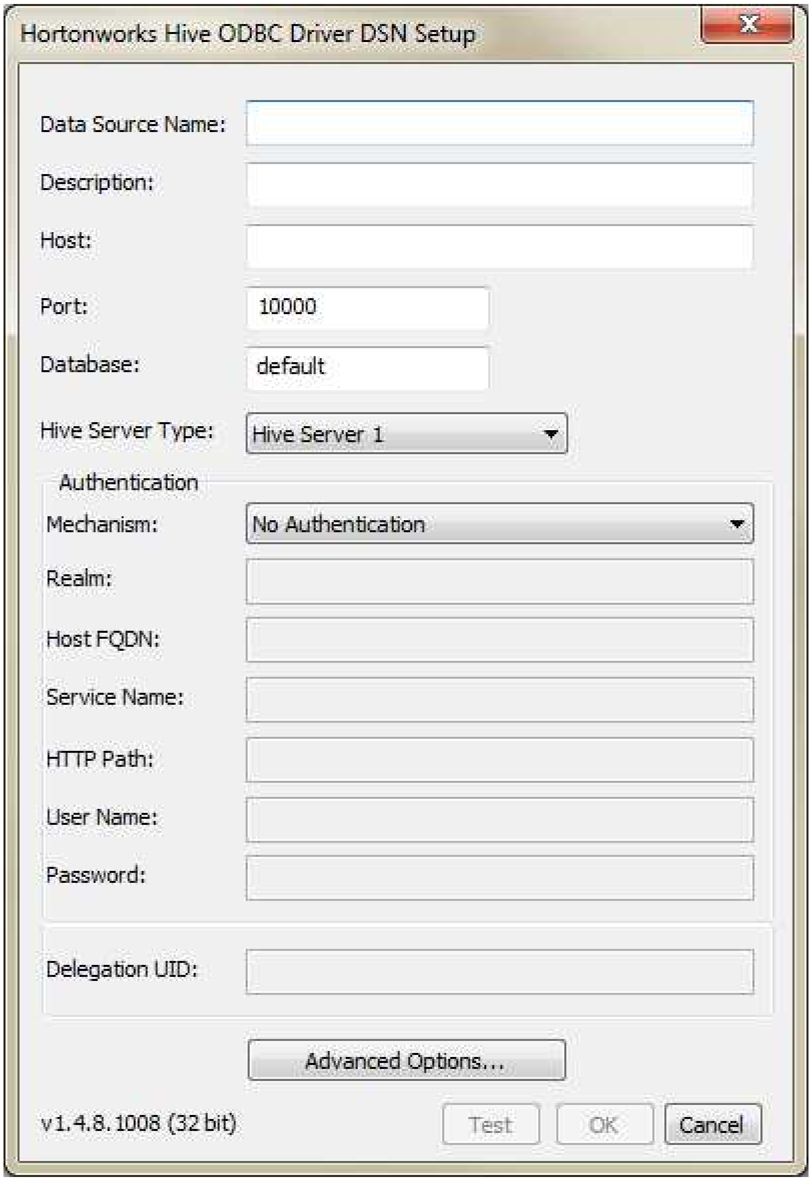
   A system DSN can be seen by all users that login to a workstation. A user DSN is specific to a user on the workstation. It can only be seen by the user who creates it.

7. Click **Add**.
   The Create New Data Source window opens.



Name	Version
Hortonworks Hive ODBC Driver	1.04.08.1008

8. Select **Hortonworks Hive ODBC Driver** and then click **Finish**. The Hortonworks Hive ODBC Driver DSN Setup window opens.



Hortonworks Hive ODBC Driver DSN Setup

Data Source Name:

Description:

Host:

Port: 10000

Database: default

Hive Server Type: Hive Server 1

Authentication

Mechanism: No Authentication

Realm:

Host FQDN:

Service Name:

HTTP Path:

User Name:

Password:

Delegation UID:

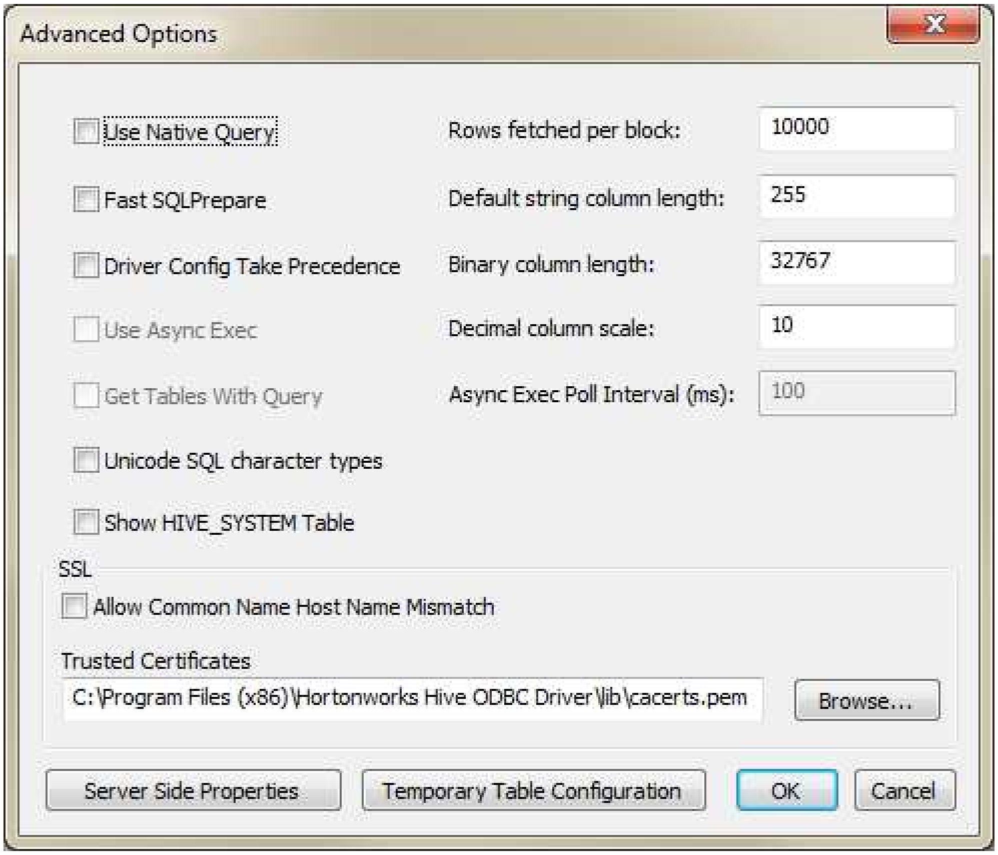
v1.4.8.1008 (32 bit)

9. In the **Data Source Name** text box, type a name for your DSN.
10. Optionally, In the **Description** text box, enter a description.
11. In the **Host** text box, type the IP address or hostname of the Hive server.
12. In the **Port** text box, type the listening port for the service.
13. In the **Database** text box, type the name of the database schema to use when a schema is not explicitly specified in a query. Queries on other schemas can still be issued by explicitly specifying the schema in the query. To determine the appropriate database schema to use, type the `show databases` command at the Hive command prompt to inspect your databases.

14. For the **Hive Server Type**, select either **Hive Server 1** or **Hive Server 2**.

    Optionally, if you selected Hive Server 2 as the Hive server type, you can configure authentication. For detailed instructions, refer to the section, "Configure authentication".
15. Optionally, if the operations against Hive are to be done on behalf of a user that is different than the authenticated user for the connection, enter the user name of the user to be delegated in the **Delegation UID** text box.

16. Optionally, click **Advanced Options**.
    The Advanced Options window opens.



Advanced Options

□ Use Native Query

□ Fast SQLPrepare

□ Driver Config Take Precedence

□ Use Async Exec

□ Get Tables With Query

□ Unicode SQL character types

□ Show HIVE_SYSTEM Table

Rows fetched per block: 10000

Default string column length: 255

Binary column length: 32767

Decimal column scale: 10

Async Exec Poll Interval (ms): 100

SSL

□ Allow Common Name Host Name Mismatch

Trusted Certificates: C:\Program Files (x86)\Hortonworks Hive ODBC Driver\lib\cacerts.pem

a) Select the **Use Native Query** checkbox to disable the SQL Connector feature.

   Note: The SQL Connector feature has been added to the driver to apply transformations to the queries emitted by an application to convert them into an equivalent form in HiveQL. If the application is Hive aware and already emits HiveQL then turning off the SQL Connector feature avoids the extra overhead of query transformation.

b) Select the **Fast SQLPrepare** checkbox to defer query execution to SQLExecute.

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

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

d) Select the **Use Async Exec** checkbox to use the asynchronous version of the API call against Hive for executing a query.

Note: This option only take effect when connecting to Hive cluster running Hive 0.12.0 or higher.

Note: Due to the problem in Hive 0.12 reported in JIRA HIVE-5230, Hive returns generic error messages for errors that occur during query execution. To find out about what the actual error message you may turn off asynchronous query execution and execute the query again.

e) Select the **Get Tables With Query** checkbox to retrieve the names of tables in a particular database using the GET TABLES query.

Note: This setting is only applicable when connecting to Hive Server 2.

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

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

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

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

i) In the **Default string column length** field, type the default string column length to use.

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

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

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

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

l) In the **Async Exec Poll Interval** field, enter the time in millisecond between each poll of the query execution status when using asynchronous query execution.

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

m) Select the **Allow Common Name Host Name Mismatch** checkbox to accept a CA issued certificate even when the common name in the certificate doesn't match the host name of the Hive server.

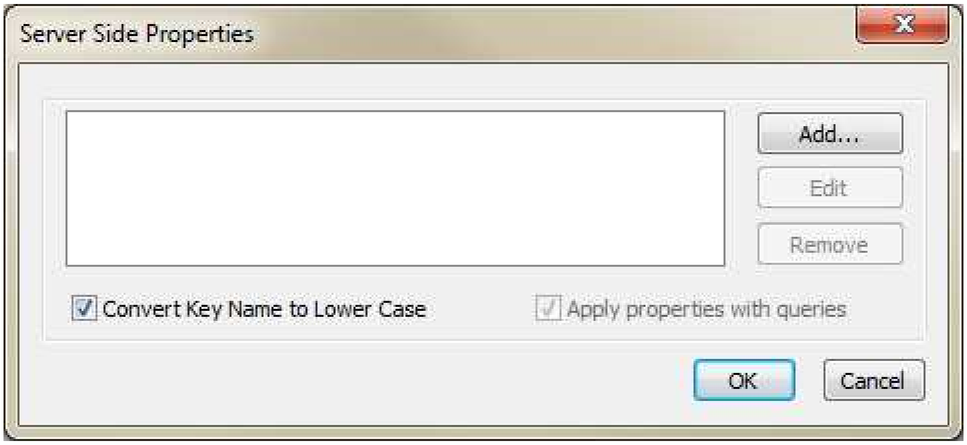
   Note: This setting is only applicable to **User Name and Password (SSL)** and **HTTPS** authentication mechanisms and will be ignored by other authentication mechanisms.

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

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

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

o) Optionally, click **Server Side Properties**. The Server Side Properties dialog opens.



☑ Convert Key Name to Lower Case

☑ Apply properties with queries

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

   OR

   To edit a server-side property, select the property to edit in the Server Side Properties area, then click the **Edit** button, then update the Key and Value fields as needed, and then click **OK**

   OR

   To delete a server-side property, select the property to remove in the Server Side Properties area, and then click the **Remove** button. In the confirmation dialog, click **Yes**

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

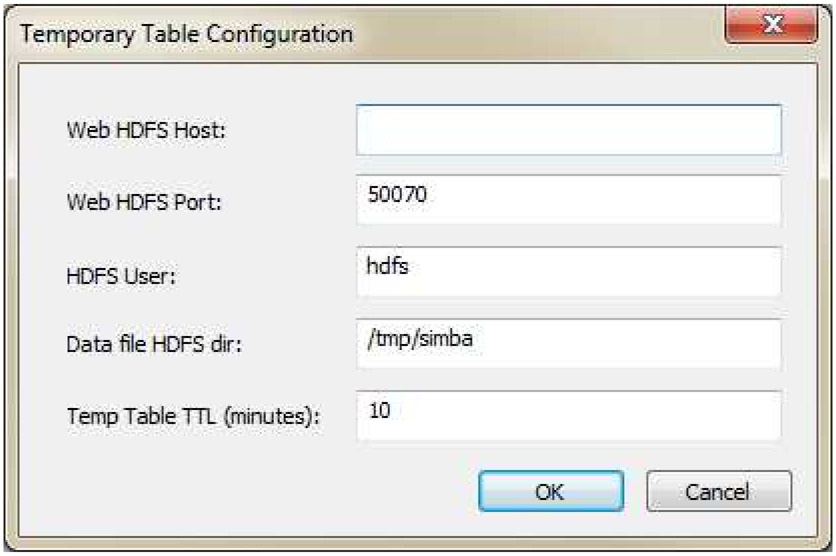
ii. If you selected Hive Server 2 as the Hive server type, then select or clear the **Apply Server Side Properties with Queries** check box as needed.

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

iii. Select the **Convert SSP Key Name to Lower Case** checkbox to force the driver to convert server side property key name to all lower case characters.

iv. Click **OK**

p) Optionally, click **Temporary Table Configuration**.


Temporary Table Configuration

Web HDFS Host:

Web HDFS Port: 50070

HDFS User: hdfs

Data file HDFS dir: /tmp/simba

Temp Table TTL (minutes): 10

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

ii. In the **Web HDFS Port** text box, enter the WebHDFS port for the namenode.

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

iv. In the **Data file HDFS dir** text box, enter the HDFS directory that the driver will use to store the necessary files for supporting the Temporary Table feature.

Note: Due to a bug in Hive (see https://issues.apache.org/jira/browse/HIVE-4554) space characters in HDFS path will not work with versions of Hive prior to 0.12.0.

v. In the **Temp Table TTL** text box, enter the number of minute a temporary table is guaranteed to exist in Hive after it is created.

vi. Click **OK**.

For details regarding the **Temporary Table** feature please refer to “Temporary Table” on page 42.

q) Click **OK**.

17. Click **Test** to test the connection and then click **OK**.

For details on configuration options available to control the behavior of Hortonworks Hive ODBC Driver using DSN, see “Appendix C: Driver Configuration Options” on page 37.

Configure authentication

Note: Authentication is only available for server of type Hive Server 2. Authentication is not available for server of type Hive Server 1.

If you are using an application that makes direct connections to Hive instead of using standard ODBC Data Sources, refer to “Appendix B: Driver Authentication Configuration for Windows”.

Unlike Hive Server 1, Hive Server 2 supports multiple authentication mechanisms. You must determine the authentication type your server is using. The authentication methods available are as follows:

- No Authentication
- Kerberos
- User Name

For **No Authentication**, no additional details are required.

For **User Name** authentication, select **User Name** in the **Mechanism** field and then type a user name in the User Name field.

For **Kerberos** authentication, Kerberos must be configured before using the driver with Kerberos authentication. Refer to “Appendix A: Configuring Kerberos Authentication for Windows”.

Note: If you installed HDP using Ambari, by default the authentication method is **User Name**.

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

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

hive.server2.authentication	**hive.server2.enable.doAs**	**Driver Authentication Mechanism**
NOSASL	False	No Authentication
KERBEROS	True or False	Kerberos
NONE	True or False	User Name

Refer to *Manual Installs* section of the HDP documentation at http://docs.hortonworks.com for a complete explanation of the authentication mechanisms.

Configuring Authentication in DSN

Using No Authentication

No additional details are required when using No Authentication.

Example connection string for Hive Server 1:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs1 host>;Port=<hs1 port>;
HiveServerType=1;AuthMech=0;Schema=<Hive database>
```

Example connection string for Hive Server 2:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=0;Schema=<Hive database>
```

Using Kerberos

Kerberos must be configured before using the driver with Kerberos authentication. Refer to “Appendix A: Configuring Kerberos Authentication for Windows”.

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

1. In the Mechanism field, select Kerberos.
2. If there is no default realm configured for your Kerberos setup, then type the value for the Kerberos realm of the HiveServer2 host. Otherwise leave it blank.

The Realm is only needed if your Kerberos setup does not define a default realm or if the realm of your HiveServer2 is not the default.

3. In the Host FQDN field, type the value for the fully qualified domain name of the HiveServer2 host.
4. In the Service Name field, type the value for the service name of the Hive Server 2. For example, if the principle for the HiveServer2 is "hive/fully.qualified.domain.name@YOUR-REALM.COM", then the value in the service name field should be hive. If you are unsure of the correct service name to use for your particular Hadoop deployment, see your Hadoop administrator.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=1;Schema=<Hive database>;KrbRealm=<Kerberos
Realm>;KrbHostFQDN=<hs2 fully qualified domain name>;KrbServiceName=<hs2
service name>
```

Using User Name

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

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=2;Schema=<Hive database>;UID=<user name>
```

Using User Name and Password

To configure your DSN for User Name and Password authentication:

1. In the Hortonworks Hive ODBC Driver DSN Setup dialog, click the drop-down arrow next to the Mechanism field, and then select User Name and Password.
2. In the User Name field, type an appropriate credential.
3. In the Password field, type the password corresponding to the user name you typed in step 2.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=3;Schema=<Hive database>;UID=<user
name>;PWD=<password>
```

Using User Name and Password (SSL)

To configure User Name and Password (SSL) authentication:

1. Click the drop-down arrow next to the Mechanism field, and then select User Name and Password (SSL).
2. In the User Name field, type an appropriate credential.
3. In the Password field, type the password corresponding to the user name you typed in step 2.

Note: SSL support in Hive Server 2 is only available starting with Hive 0.13.

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

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

Note: Optionally you can configure the Trusted Certificates setting in the Advanced Options to allow the driver to load SSL certificates from the specified file. The driver will default to using the trusted CA certificates PEM file install with the driver if this setting is not set.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=4;Schema=<Hive database>;UID=<user
name>;PWD=<password>
```

Using Windows Azure HDInsight Emulator

To connect to Hive server on Microsoft's Windows Azure HDInsight Emulator:

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

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<HDInsight Emulator
host>;Port=<HDInsight Emulator port>; HiveServerType=2;AuthMech=5;Schema=<Hive
database>;UID=<user name>;PWD=<password>;HTTPPath=<hs2 HTTP path>
```

Using Windows Azure HDInsight Service

To connect to Hive server on Microsoft's Windows Azure HDInsight Service:

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

Note: The driver doesn't allow self-signed SSL certificate for this authentication mechanism.

Note: The common name of the CA issued certificate must match the host name of the Hive Server.

Note: Optionally you can configure the Trusted Certificates setting in the Advanced Options to allow the driver to load SSL certificates from the specified file. The driver will default to using the trusted CA certificates PEM file install with the driver if this setting is not set.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<Azure HDInsight Service
host>;Port=443; HiveServerType=2;AuthMech=6;Schema=<Hive database>;UID=<user
name>;PWD=<password>;HTTPPath=<hs2 HTTP path>
```

Using HTTP

To configure HTTP authentication:

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

Note: HTTP support in Hive Server 2 is only available starting with Hive 0.13.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=7;Schema=<Hive database>;UID=<user
name>;PWD=<password>;HTTPPath=<hs2 HTTP path>
```

Using HTTPS

To use HTTPS authentication:

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

Note: HTTPS support in Hive Server 2 is only available starting with Hive 0.13.

Note: The HTTPS authentication method can also be used to connect to Hive Server 2 via the Knox gateway. Please refer to the Knox documentation to determine what user credentials to use and what value to set for HTTPPath.

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

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

Note: Optionally you can configure the Trusted Certificates setting in the Advanced Options to allow the driver to load SSL certificates from the specified file. The driver will default to using the trusted CA certificates PEM file install with the driver if this setting is not set.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=8;Schema=<Hive database>;UID=<user
name>;PWD=<password>;HTTPPath=<hs2 HTTP path>
```

Using Kerberos over HTTP

To configure Kerberos over HTTP authentication:

1. In the Hortonworks Hive ODBC Driver DSN Setup dialog, click the drop-down arrow next to the Mechanism field in the Authentication area, and then select Kerberos over HTTP.
2. If there is no default realm configured for your Kerberos setup, then type the value for the Kerberos realm of the HiveServer2 host. Otherwise leave it blank. The Realm is only needed if your Kerberos setup does not define a default realm or if the realm of your HiveServer2 is not the default.
3. In the Host FQDN field, type the value for the fully qualified domain name of the HiveServer2 host.

4. In the Service Name field, type the value for the service name of the Hive Server 2. For example, if the principle for the HiveServer2 is "hive/fully.qualified.domain.name@YOUR-REALM.COM", then the value in the service name field should be hive. If you are unsure of the correct service name to use for your particular Hadoop deployment, see your Hadoop administrator.
5. In the HTTP Path field, type the partial URL corresponding to the Hive server.

   Note: Kerberos over HTTP support in Hive Server 2 is only available starting with Hive 0.13.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=9;Schema=<Hive database>;KrbRealm=<Kerberos
Realm>;KrbHostFQDN=<hs2 fully qualified domain name>;KrbServiceName=<hs2
service name>;HTTPPath=<hs2 HTTP path>
```

Using Kerberos over HTTPS

To use Kerberos over HTTPS authentication:

1. In the Hortonworks Hive ODBC Driver DSN Setup dialog, click the drop-down arrow next to the Mechanism field in the Authentication area, and then select Kerberos over HTTPS.
2. If there is no default realm configured for your Kerberos setup, then type the value for the Kerberos realm of the HiveServer2 host. Otherwise leave it blank. The Realm is only needed if your Kerberos setup does not define a default realm or if the realm of your HiveServer2 is not the default.
3. In the Host FQDN field, type the value for the fully qualified domain name of the HiveServer2 host.
4. In the Service Name field, type the value for the service name of the Hive Server 2. For example, if the principle for the HiveServer2 is "hive/fully.qualified.domain.name@YOUR-REALM.COM", then the value in the service name field should be hive. If you are unsure of the correct service name to use for your particular Hadoop deployment, see your Hadoop administrator.
5. In the HTTP Path field, type the partial URL corresponding to the Hive server.

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

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

   Note: Optionally you can configure the Trusted Certificates setting in the Advanced Options to allow the driver to load SSL certificates from the specified file. The driver will default to using the trusted CA certificates PEM file install with the driver if this setting is not set.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=10;Schema=<Hive database>;KrbRealm=<Kerberos
Realm>;KrbHostFQDN=<hs2 fully qualified domain name>;KrbServiceName=<hs2
service name>;HTTPPath=<hs2 HTTP path>
```

Linux Driver

System Requirements

- Red Hat® Enterprise Linux® (RHEL) 5.0, CentOS 5.0 or SUSE Linux Enterprise Server (SLES) 11. Both 32 and 64-bit editions are supported.
- 45 MB of available disk space.
- An installed ODBC Driver Manager, for example:
  - iODBC 3.52.7 or above
  - unixODBC 2.3.0 or above

The Hortonworks Hive ODBC Driver with SQL Connector requires a Hadoop cluster with the Hive service installed and running.

The Hortonworks Hive ODBC Driver with SQL Connector is suitable for use with all versions of Hive.

Installation

There are two versions of the driver for Linux:

- **hive-odbc-native-32bit-*<version>*-*<release>*.i686.rpm** for 32-bit
- **hive-odbc-native-*<version>*-*<release>*.x86_64.rpm** for 64-bit

Please refer to your Linux distribution's documentation for instructions on how to install RPM packages.

The version of the driver that you select should match the bitness of the application. For example, if the 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. It is allowable to install both versions of the driver.

Driver Directories

The Hortonworks Hive ODBC Driver files are installed in the following directories:

- /usr/lib/hive/lib/native/hiveodbc/ErrorMessages – Error messages files directory
- /usr/lib/hive/lib/native/hiveodbc/Setup – Sample configuration files directory
- /usr/lib/hive/lib/native/Linux-i386-32 – 32-bit shared libraries directory
- /usr/lib/hive/lib/native/Linux-amd64-64 – 64-bit shared libraries directory

Configuration

ODBC Configuration Files

ODBC driver managers use configuration files to define and configure ODBC data sources and drivers. By default, the configuration files reside in the user's home directory. The configuration files are:

- **.odbc.ini** – The file used to define ODBC data sources (required)
- **.odbcinst.ini** – The file used to define ODBC drivers (optional)
- **.hortonworks.hiveodbc.ini** – The file used to configure the Hortonworks Hive ODBC Driver (required)

Sample ODBC Configuration Files

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

- **odbc.ini**
- **odbcinst.ini**
- **hortonworks.hiveodbc.ini**

The names of the sample configuration files do not begin with a period (.) so that they will appear in normal directory listings. A filename beginning with a period (.) is hidden. For **odbc.ini** and **odbcinst.ini**, if the default location is used, the filenames must begin with a period (.). For **hortonworks.hiveodbc.ini**, the filename must begin with a period (.) and must reside in the user's home directory.

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

ODBCINI and ODBCSYSINI Environment Configuration

By default, the configuration files reside in the user's home directory. However, two environment variables, **ODBCINI** and **ODBCSYSINI**, can be used to specify an alternative location of the **.odbc.ini** and **.odbcinst.ini** configuration files. For example, in the Bash shell, the location could be specified as follows:

```
export ODBCINI=/usr/local/odbc/myodbc.ini
```

```
export ODBCSYSINI=/usr/local/odbc/myodbcinst.ini
```

Refer to your Linux shell documentation for the exact syntax for setting environment variables.

ODBC Data Source Configuration File Overview

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

- [ODBC]
  The [ODBC] section is used to control global ODBC configuration such as ODBC tracing.

- [ODBC Data Sources]
  The [ODBC Data Sources] section is used to specify the available data sources.
- Data Source definitions ([<data source name>])
  The Data Source definitions are used to define the actual data source configurations.

For example, an **.odbc.ini** configuration file might look something like this:

```
[ODBC]
InstallDir=/usr/local/odbc

[ODBC Data Sources]
Sample Hortonworks Hive DSN 32=Hortonworks Hive ODBC Driver 32-bit

[Sample Hortonworks Hive DSN 32]
Driver=/usr/lib/hive/lib/native/Linux-i386-32/libhortonworkshiveodbc32.so
HOST=myhiveserver
PORT=10000
```

Create a Data Source

To create a data source:

1. Open the **.odbc.ini** configuration file in a text editor.
2. Add a new entry to the [ODBC Data Sources] section. Type the data source name (DSN) and the driver name. It might look something like this:

   ```
   Sample Hortonworks Hive DSN 32=Hortonworks Hive ODBC Driver 32-bit
   ```
3. Add a new section with a name that matches the data source name (DSN). This section will contain the configuration options. They are specified as key-value pairs. For example, it might look something like this:

   ```
   [Sample Hortonworks Hive DSN 32]
   Driver=/usr/lib/hive/lib/native/Linux-i386-32/libhortonworkshiveodbc32.so
   HOST=myhiveserver
   PORT=10000
   ```
4. Save the **.odbc.ini** configuration file.

For details on configuration options available to control the behavior of Hortonworks Hive ODBC Driver using DSN, see “Appendix C: Driver Configuration Options” on page 37.

For details regarding the **Temporary Table** feature please refer to “Temporary Table” on page 42.

Configuring Authentication

Using No Authentication

To use no authentication:

1. Set the AuthMech configuration key for the DSN to 0

Example connection string for Hive Server 1:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs1 host>;Port=<hs1 port>;
HiveServerType=1;AuthMech=0;Schema=<Hive database>
```

Example connection string for Hive Server 2:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=0;Schema=<Hive database>
```

Using Kerberos

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

To configure a DSN using Hortonworks ODBC Driver with SQL Connector for Apache Hive to use Kerberos authentication:

1. Set the AuthMech configuration key for the DSN to 1
2. If your Kerberos setup does not define a default realm or if the realm of your Hive server is not the default, then set the appropriate realm using the KrbRealm key.
3. Set the KrbHostFQDN key to the fully qualified domain name of the Hive Server 2 host.
4. Set the KrbServiceName key to the service name of the Hive Server 2.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=1;Schema=<Hive database>;KrbRealm=<Kerberos
Realm>;KrbHostFQDN=<hs2 fully qualified domain name>;KrbServiceName=<hs2
service name>
```

Using User Name

To configure User Name authentication:

1. Set the AuthMech configuration key for the DSN to 2
2. Set the UID key to the appropriate user name recognized by the Hive server.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=2;Schema=<Hive database>;UID=<user name>
```

Using User Name and Password

To configure User Name and Password authentication:

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

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=3;Schema=<Hive database>;UID=<user
name>;PWD=<password>
```

Using User Name and Password (SSL)

To configure User Name and Password (SSL) authentication:

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

Note: SSL support in Hive Server 2 is only available starting with Hive 0.13.

Note: The driver always accepts the use of self-signed SSL certificate.

Note: Optionally you can configure the CAIssuedCertNamesMismatch setting to control whether the driver allows the common name of a CA issued certificate to not match the host name of the Hive server. For self-signed certificates the driver always allow common name of the certificate to not match the host name. See “Appendix C: Driver Configuration Options” on page 49.

Note: Optionally you can configure the TrustedCerts setting in the Advanced Options to allow the driver to load SSL certificates from the specified file. The driver will default to using the trusted CA certificates PEM file install with the driver if this setting is not set. See “Appendix C: Driver Configuration Options” on page 49.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=4;Schema=<Hive database>;UID=<user
name>;PWD=<password>
```

Using Windows Azure HDInsight Emulator

To connect to Hive server on Microsoft’s Windows Azure HDInsight Emulator :

1. Set the AuthMech configuration key for the DSN to 5

2. Set the HTTPPath key to the partial URL corresponding to the Hive server on Windows Azure HDInsight Emulator.
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.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<HDInsight Emulator
host>;Port=<HDInsight Emulator port>; HiveServerType=2;AuthMech=5;Schema=<Hive
database>;UID=<user name>;PWD=<password>;HTTPPath=<hs2 HTTP path>
```

Using Windows Azure HDInsight Service

To connect to Hive server on Microsoft's Windows Azure HDInsight Service:

1. Set the AuthMech configuration key for the DSN to 6
2. Set the HTTPPath key to the partial URL corresponding to the Hive server.
3. Set the UID key to an appropriate user name for accessing the Hive server.
4. Set the PWD key to the password corresponding to the user name you typed in step 3.

Note: The driver doesn't allow self-signed SSL certificate for this authentication mechanism.

Note: The driver doesn't allow the common name of the SSL certificate to not match the hostname of the Hive server.

Note: Optionally you can configure the TrustedCerts setting in the Advanced Options to allow the driver to load SSL certificates from the specified file. The driver will default to using the trusted CA certificates PEM file install with the driver if this setting is not set. See "Appendix C: Driver Configuration Options for Linux and Mac OS X" on page 28.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<Azure HDInsight Service
host>;Port=443; HiveServerType=2;AuthMech=6;Schema=<Hive database>;UID=<user
name>;PWD=<password>;HTTPPath=<hs2 HTTP path>
```

Using HTTP

To configure HTTP authentication:

1. Set the AuthMech configuration key for the DSN to 7
2. Set the HTTPPath key to the partial URL corresponding to the Hive server.
3. Set the UID key to an appropriate user name for accessing the Hive server.

4. Set the PWD key to the password corresponding to the user name you typed in step 3.

Note: HTTP support in Hive Server 2 is only available starting with Hive 0.13.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=7;Schema=<Hive database>;UID=<user
name>;PWD=<password>;HTTPPath=<hs2 HTTP path>
```

Using HTTPS

To configure HTTPS authentication:

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

Note: HTTPS support in Hive Server 2 is only available starting with Hive 0.13.

Note: The HTTPS authentication method can also be used to connect to Hive Server 2 via the Knox gateway. Please refer to the Knox documentation to determine what user credentials to use and what value to set for HTTPPath.

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

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

Note: Optionally you can configure the Trusted Certificates setting in the Advanced Options to allow the driver to load SSL certificates from the specified file. The driver will default to using the trusted CA certificates PEM file install with the driver if this setting is not set.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=8;Schema=<Hive database>;UID=<user
name>;PWD=<password>;HTTPPath=<hs2 HTTP path>
```

Using Kerberos over HTTP

To configure Kerberos over HTTP authentication:

1. Set the AuthMech configuration key for the DSN to 9.

2. If your Kerberos setup does not define a default realm or if the realm of your Hive server is not the default, then set the appropriate realm using the KrbRealm key.
3. Set the KrbHostFQDN key to the fully qualified domain name of the Hive Server 2 host.
4. Set the KrbServiceName key to the service name of the Hive Server 2.
5. Set the HTTPPath key to the partial URL corresponding to the Hive server.

Note: Kerberos over HTTP support in Hive Server 2 is only available starting with Hive 0.13.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=9;Schema=<Hive database>;KrbRealm=<Kerberos
Realm>;KrbHostFQDN=<hs2 fully qualified domain name>;KrbServiceName=<hs2
service name>;HTTPPath=<hs2 HTTP path>
```

Using Kerberos over HTTPS

To configure Kerberos over HTTPS authentication:

1. Set the AuthMech configuration key for the DSN to 10.
2. If your Kerberos setup does not define a default realm or if the realm of your Hive server is not the default, then set the appropriate realm using the KrbRealm key.
3. Set the KrbHostFQDN key to the fully qualified domain name of the Hive Server 2 host.
4. Set the KrbServiceName key to the service name of the Hive Server 2.
5. Set the HTTPPath key to the partial URL corresponding to the Hive server.

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

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

Note: Optionally you can configure the Trusted Certificates setting in the Advanced Options to allow the driver to load SSL certificates from the specified file. The driver will default to using the trusted CA certificates PEM file install with the driver if this setting is not set.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=10;Schema=<Hive database>;KrbRealm=<Kerberos
Realm>;KrbHostFQDN=<hs2 fully qualified domain name>;KrbServiceName=<hs2
service name>;HTTPPath=<hs2 HTTP path>
```

ODBC Drivers Configuration File Overview

ODBC Drivers are defined in the **.odbcinst.ini** configuration file. This configuration is optional because drivers can be specified directly in the **.odbc.ini** configuration file as discussed in the previous section.

The file is divided into these sections:

- [ODBC Drivers]
  The [ODBC Drivers] section is used to specify the available drivers.
- Driver definitions ([<driver name>])
  The Driver definitions are used to define the actual driver configurations.

For example, an **.odbcinst.ini** configuration file might look something like this:

```
[ODBC Drivers]

Hortonworks Hive ODBC Driver 32-bit=Installed

Hortonworks Hive ODBC Driver 64-bit=Installed


[Hortonworks Hive ODBC Driver 32-bit]

Driver=/usr/lib/hive/lib/native/Linux-i386-32/libhortonworkshiveodbc32.so

Description=Hortonworks Hive ODBC Driver (32-bit)


[Hortonworks Hive ODBC Driver 64-bit]

Driver=/usr/lib/hive/lib/native/Linux-amd64-64/libhortonworkshiveodbc64.so

Description=Hortonworks Hive ODBC Driver (64-bit)
```

Define a Driver

To define a driver:

1. Open the **.odbcinst.ini** configuration file in a text editor.
2. Add a new entry to the [ODBC Drivers] section. Type driver name and the value “Installed”. This driver name should be used for the “Driver” value in the data source definition instead of the driver shared library name.

   For example, it might look something like this:

   ```
   Hortonworks Hive ODBC Driver 32-bit=Installed
   ```
3. Add a new section with a name that matches the new driver name. This section will contain the configuration options. They are specified as key-value pairs. For example, it might look something like this:

   ```
   [Hortonworks Hive ODBC Driver 32-bit]

   Driver=/usr/lib/hive/lib/native/Linux-i386-32/libhortonworkshiveodbc32.so

   Description=Hortonworks Hive ODBC Driver (32-bit)
   ```
4. Save the **.odbcinst.ini** configuration file.

Configure the Hortonworks Hive ODBC Driver

To configure the Hortonworks Hive ODBC Driver to work with your ODBC Driver Manager:

1. Open the **.hortonworks.hiveodbc.ini** configuration file in a text editor.
2. Edit the DriverManagerEncoding setting.
   This setting is usually set to **UTF-16** or **UTF-32** depending on the ODBC Driver Manager being used. iODBC uses **UTF-32** and unixODBC uses **UTF-16**. Consult your ODBC Driver Manager documentation for the correct setting to use.
3. Edit the ODBCInstLib setting.
   This setting is set to the ODBCInst shared library for the ODBC Driver Manager being used. The configuration file defaults to iODBC's **libiodbcinst.so** shared library. You can specify the absolute or relative filename for the library. If you intend to use the relative filename for the library, the path to the library must be included in your **LD_LIBRARY_PATH** setting. Consult your ODBC Driver Manager documentation for the correct library to use.
4. Save the **.hortonworks.hiveodbc.ini** configuration file.

Configure the Library Path

In the ODBC configuration files, the driver libraries can be specified using absolute or relative paths. If relative paths are desired, set **LD_LIBRARY_PATH** to include:

- /usr/lib/hive/lib/native/Linux-i386-32
- /usr/lib/hive/lib/native/Linux-amd64-64

Refer to your Linux shell documentation for the exact syntax for setting environment variables.

Configure Kerberos Authentication

For more information about how to configure Kerberos authentication, refer to the documentation for your operating system.

Note: Authentication is not available for the server type of Hive Server 1.

Mac OS X Driver

System Requirements

- Mac OS X version 10.6.8 or later.
- 86 MB of available disk space.
- An installed ODBC Driver Manager, for example:
  - iODBC 3.52.7 or above

The Hortonworks Hive ODBC Driver with SQL Connector requires a Hadoop cluster with the Hive service installed and running.

The Hortonworks Hive ODBC Driver with SQL Connector is suitable for use with all versions of Hive and the driver works with both 32 and 64-bit applications.

Installation

1. Double click the `hive-odbc-native.dmg` file.
   The Hortonworks Hive ODBC Driver volume is mounted.
2. Double click the `hive-odbc-native.pkg` file.
   The installer opens.
3. Follow the instructions in the installer and when it has finished installing, click Close.

Driver Directories

The Hortonworks Hive ODBC Driver files are installed in the following directories:

- `/usr/lib/hive/lib/native/hiveodbc/ErrorMessages` – Error messages files directory
- `/usr/lib/hive/lib/native/hiveodbc/Setup` – Sample configuration files directory
- `/usr/lib/hive/lib/native/universal` – Binaries directory

Configuration

ODBC Configuration Files

ODBC driver managers use configuration files to define and configure ODBC data sources and drivers. By default, the configuration files reside in the user's home directory. The configuration files are:

- **.odbc.ini** – The file used to define ODBC data sources (required)
- **.odbcinst.ini** – The file used to define ODBC drivers (optional)
- **.hortonworks.hiveodbc.ini** – The file used to configure the Hortonworks Hive ODBC Driver (required)

Sample ODBC Configuration Files

The driver installation contains the following sample configuration files in the `/usr/lib/hive/lib/native/hiveodbc/Setup` directory:

- **odbc.ini**
- **odbcinst.ini**
- **hortonworks.hiveodbc.ini**

The names of the sample configuration files do not begin with a period (.) so that they will appear in normal directory listings. A filename beginning with a period (.) is hidden. For **odbc.ini** and **odbcinst.ini**, if the default location is used, the filenames must begin with a period (.). For **hortonworks.hiveodbc.ini**, the filename must begin with a period (.) and must reside in the user's home directory.

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

ODBCINI and ODBCSYSINI Environment Configuration

By default, the configuration files reside in the user's home directory. However, two environment variables, **ODBCINI** and **ODBCSYSINI**, can be used to specify an alternative location of the **.odbc.ini** and **.odbcinst.ini** configuration files. For example, in the Bash shell, the location could be specified as follows:

```
export ODBCINI=/usr/local/odbc/myodbc.ini

export ODBCSYSINI=/usr/local/odbc/myodbcinst.ini
```

ODBC Data Source Configuration File Overview

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

- [ODBC]
  The [ODBC] section is used to control global ODBC configuration such as ODBC tracing.
- [ODBC Data Sources]
  The [ODBC Data Sources] section is used to specify the available data sources.
- Data Source definitions ([<data source name>])
  The Data Source definitions are used to define the actual data source configurations.

For example, an **.odbc.ini** configuration file might look something like this:

```
[ODBC]

InstallDir=/usr/local/odbc


[ODBC Data Sources]

Hortonworks Hive DSN=Hortonworks Hive ODBC Driver


[Hortonworks Hive DSN]

Driver=/usr/lib/hive/lib/native/universal/libhortonworkshiveodbc.dylib

HOST=myhiveserver

PORT=10000
```

Create a Data Source

To create a data source:

1. Open the **.odbc.ini** configuration file in a text editor.
2. Add a new entry to the [ODBC Data Sources] section. Type the data source name (DSN) and the driver name. It might look something like this:

   ```
   Hortonworks Hive DSN=Hortonworks Hive ODBC Driver
   ```

3. Add a new section with a name that matches the data source name (DSN). This section will contain the configuration options. They are specified as key-value pairs. For example, it might look something like this:

   ```
   [Hortonworks Hive DSN]
   Driver=/usr/lib/hive/lib/native/universal/libhortonworkshiveodbc.dylib
   HOST=myhiveserver
   PORT=10000
   ```

4. Save the **.odbc.ini** configuration file.

For details on configuration options available to control the behavior of Hortonworks Hive ODBC Driver using DSN, see “Appendix C: Driver Configuration Options” on page 49.

For details regarding the **Temporary Table** feature please refer to “Temporary Table” on page 42.

Configuring Authentication

Using No Authentication

To use no authentication:

1. Set the AuthMech configuration key for the DSN to 0

Example connection string for Hive Server 1:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs1 host>;Port=<hs1 port>;
HiveServerType=1;AuthMech=0;Schema=<Hive database>
```

Example connection string for Hive Server 2:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=0;Schema=<Hive database>
```

Using Kerberos

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

To configure a DSN using Hortonworks ODBC Driver with SQL Connector for Apache Hive to use Kerberos authentication:

1. Set the AuthMech configuration key for the DSN to 1
2. If your Kerberos setup does not define a default realm or if the realm of your Hive server is not the default, then set the appropriate realm using the KrbRealm key.
3. Set the KrbHostFQDN key to the fully qualified domain name of the Hive Server 2 host.
4. Set the KrbServiceName key to the service name of the Hive Server 2.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=1;Schema=<Hive database>;KrbRealm=<Kerberos
```

```
Realm>;KrbHostFQDN=<hs2 fully qualified domain name>;KrbServiceName=<hs2
service name>
```

Using User Name

To configure User Name authentication:

1. Set the AuthMech configuration key for the DSN to 2
2. Set the UID key to the appropriate user name recognized by the Hive server.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=2;Schema=<Hive database>;UID=<user name>
```

Using User Name and Password

To configure User Name and Password authentication:

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

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=3;Schema=<Hive database>;UID=<user
name>;PWD=<password>
```

Using User Name and Password (SSL)

To configure User Name and Password (SSL) authentication:

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

Note: SSL support in Hive Server 2 is only available starting with Hive 0.13.

Note: The driver always accepts the use of self-signed SSL certificate.

Note: Optionally you can configure the CAIssuedCertNamesMismatch setting to control whether the driver allows the common name of a CA issued certificate to not match the host name of the Hive server. For self-signed certificates the driver always allow common name of the certificate to not match the host name. See “Appendix C: Driver Configuration Options” on page 49.

Note: Optionally you can configure the TrustedCerts setting in the Advanced Options to allow the driver to load SSL certificates from the specified file. The driver will default to using the trusted CA certificates PEM file install with the driver if this setting is not set. See “Appendix C: Driver Configuration Options” on page 49.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=4;Schema=<Hive database>;UID=<user
name>;PWD=<password>
```

Using Windows Azure HDInsight Emulator

To connect to Hive server on Microsoft’s Windows Azure HDInsight Emulator :

1. Set the AuthMech configuration key for the DSN to 5
2. Set the HTTPPath key to the partial URL corresponding to the Hive server on Windows Azure HDInsight Emulator.
3. Set the UID key to an appropriate user name for accessing the Hive server.
4. Set the PWD key to the password corresponding to the user name you typed in step 3.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<HDInsight Emulator
host>;Port=<HDInsight Emulator port>; HiveServerType=2;AuthMech=5;Schema=<Hive
database>;UID=<user name>;PWD=<password>;HTTPPath=<hs2 HTTP path>
```

Using Windows Azure HDInsight Service

To connect to Hive server on Microsoft’s Windows Azure HDInsight Service:

1. Set the AuthMech configuration key for the DSN to 6
2. Set the HTTPPath key to the partial URL corresponding to the Hive server.
3. Set the UID key to an appropriate user name for accessing the Hive server.
4. Set the PWD key to the password corresponding to the user name you typed in step 3.

Note: The driver doesn’t allow self-signed SSL certificate for this authentication mechanism.

Note: The driver doesn’t allow the common name of the SSL certificate to not match the hostname of the Hive server.

Note: Optionally you can configure the TrustedCerts setting in the Advanced Options to allow the driver to load SSL certificates from the specified file. The driver will default to using the trusted CA certificates PEM file install with the driver if this setting is not set. See “Appendix C: Driver Configuration Options for Linux and Mac OS X” on page 28.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<Azure HDInsight Service
host>;Port=443; HiveServerType=2;AuthMech=6;Schema=<Hive database>;UID=<user
name>;PWD=<password>;HTTPPath=<hs2 HTTP path>
```

Using HTTP

To configure HTTP authentication:

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

Note: HTTP support in Hive Server 2 is only available starting with Hive 0.13.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=7;Schema=<Hive database>;UID=<user
name>;PWD=<password>;HTTPPath=<hs2 HTTP path>
```

Using HTTPS

To configure HTTPS authentication:

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

Note: HTTPS support in Hive Server 2 is only available starting with Hive 0.13.

Note: The HTTPS authentication method can also be used to connect to Hive Server 2 via the Knox gateway. Please refer to the Knox documentation to determine what user credentials to use and what value to set for HTTPPath.

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

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

Note: Optionally you can configure the Trusted Certificates setting in the Advanced Options to allow the driver to load SSL certificates from the specified file. The driver will default to using the trusted CA certificates PEM file install with the driver if this setting is not set.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=8;Schema=<Hive database>;UID=<user
name>;PWD=<password>;HTTPPath=<hs2 HTTP path>
```

Using Kerberos over HTTP

To configure Kerberos over HTTP authentication:

6. Set the AuthMech configuration key for the DSN to 9.
7. If your Kerberos setup does not define a default realm or if the realm of your Hive server is not the default, then set the appropriate realm using the KrbRealm key.
8. Set the KrbHostFQDN key to the fully qualified domain name of the Hive Server 2 host.
9. Set the KrbServiceName key to the service name of the Hive Server 2.
10. Set the HTTPPath key to the partial URL corresponding to the Hive server.

Note: Kerberos over HTTP support in Hive Server 2 is only available starting with Hive 0.13.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=9;Schema=<Hive database>;KrbRealm=<Kerberos
Realm>;KrbHostFQDN=<hs2 fully qualified domain name>;KrbServiceName=<hs2
service name>;HTTPPath=<hs2 HTTP path>
```

Using Kerberos over HTTPS

To configure Kerberos over HTTPS authentication:

6. Set the AuthMech configuration key for the DSN to 10.
7. If your Kerberos setup does not define a default realm or if the realm of your Hive server is not the default, then set the appropriate realm using the KrbRealm key.
8. Set the KrbHostFQDN key to the fully qualified domain name of the Hive Server 2 host.
9. Set the KrbServiceName key to the service name of the Hive Server 2.
10. Set the HTTPPath key to the partial URL corresponding to the Hive server.

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

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

Note: Optionally you can configure the Trusted Certificates setting in the Advanced Options to allow the driver to load SSL certificates from the specified file. The driver will default to using the trusted CA certificates PEM file install with the driver if this setting is not set.

Example connection string:

```
Driver=Hortonworks Hive ODBC Driver;Host=<hs2 host>;Port=<hs2 port>;
HiveServerType=2;AuthMech=10;Schema=<Hive database>;KrbRealm=<Kerberos
Realm>;KrbHostFQDN=<hs2 fully qualified domain name>;KrbServiceName=<hs2
service name>;HTTPPath=<hs2 HTTP path>
```

ODBC Drivers Configuration File Overview

ODBC Drivers are defined in the **.odbcinst.ini** configuration file. This configuration is optional because drivers can be specified directly in the **.odbc.ini** configuration file as discussed in the previous section.

The file is divided into these sections:

- [ODBC Drivers]
  The [ODBC Drivers] section is used to specify the available drivers.
- Driver definitions ([<driver name>])
  The Driver definitions are used to define the actual driver configurations.

For example, an **.odbcinst.ini** configuration file might look something like this:

```
[ODBC Drivers]

Hortonworks Hive ODBC Driver=Installed


[Hortonworks Hive ODBC Driver]

Driver=/usr/lib/hive/lib/native/universal/libhortonworkshiveodbc.dylib

Description=Hortonworks Hive ODBC Driver
```

Define a Driver

To define a driver:

1. Open the **.odbcinst.ini** configuration file in a text editor.
2. Add a new entry to the [ODBC Drivers] section. Type driver name and the value “Installed”. This driver name should be used for the “Driver” value in the data source definition instead of the driver shared library name.

   For example, it might look something like this:

   ```
   Hortonworks Hive ODBC Driver=Installed
   ```

3. Add a new section with a name that matches the new driver name. This section will contain the configuration options. They are specified as key-value pairs. For example, it might look something like this:

   ```
   [Hortonworks Hive ODBC Driver]

   Driver=/usr/lib/hive/lib/native/universal/libhortonworkshiveodbc.dylib

   Description=Hortonworks Hive ODBC Driver
   ```

4. Save the **.odbcinst.ini** configuration file.

Configure the Hortonworks Hive ODBC Driver

To configure the Hortonworks Hive ODBC Driver to work with your ODBC Driver Manager:

1. Open the **.hortonworks.hiveodbc.ini** configuration file in a text editor.
2. Edit the DriverManagerEncoding setting.
   This setting is usually set to **UTF-16** or **UTF-32** depending on the ODBC Driver Manager being used. iODBC uses **UTF-32**. Consult your ODBC Driver Manager documentation for the correct setting to use.
3. Edit the ODBCInstLib setting.
   This setting is set to the ODBCInst shared library for the ODBC Driver Manager being used. The configuration file defaults to iODBC's **libiodbcinst.dylib** shared library. You can specify the absolute or relative filename for the library. If you intend to use the relative filename for the library, the path to the library must be included in your **DYLD_LIBRARY_PATH** setting. Consult your ODBC Driver Manager documentation for the correct library to use.
4. Save the **.hortonworks.hiveodbc.ini** configuration file.

Configure the Library Path

In the ODBC configuration files, the driver libraries can be specified using absolute or relative paths. If relative paths are desired, set **DYLD_LIBRARY_PATH** to include `/usr/lib/hive/lib/native/universal`.

Configure Kerberos Authentication

For more information about how to configure Kerberos authentication, refer to the documentation for your operating system.

Note: Authentication is not available for the server type of Hive Server 1.

Features

SQL Query versus HiveQL Query

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

SQL Connector

To bridge the difference between SQL and HiveQL, we have added the **SQL Connector** feature to translate standard SQL-92 queries into equivalent HiveQL queries. The **SQL Connector** performs syntactical translations and structural transformations. For example:

1. Quoted Identifiers

   HiveQL uses back-quote while SQL uses double quote when quoting identifiers. Even when a driver reports the back-quote as the quote character, some applications still generate double quoted identifiers.

2. Table Aliases

   HiveQL does not support the AS keyword between a table reference and its alias.

3. JOIN, INNER JOIN and CROSS JOIN

   SQL INNER JOIN and CROSS JOIN syntax is translated to HiveQL JOIN syntax.

4. TOP N/LIMIT

   SQL TOP N queries are transformed to HiveQL LIMIT queries.

Data Types

The following data types are supported: TINYINT, SMALLINT, INT, BIGINT, FLOAT, DOUBLE, BOOLEAN, STRING, TIMESTAMP, DATE, VARCHAR(n), DATE, DECIMAL(p,s), and CHAR(n).

The aggregate types (ARRAY, MAP and STRUCT) are not yet supported.

Catalog and Schema Support

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

Hive System Table

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

```
SELECT * FROM HIVE_SYSTEM.HIVE_SYSTEM WHERE ENVKEY LIKE ‘%hive%’
```

will return all of the Hive system environment entries whose key has the word “hive” in it. A special query, “set –v”, has to be executed to fetch this information and this is not supported by all Hive versions. For versions of Hive that do not support this type of query, the driver will return an empty result set.

Server-side Properties

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

For details on setting server-side properties for a DSN using the Windows driver, see the section “Configuration” on page 6. For details related to the Linux driver, see “Create a Data Source” on page 24. For details related to the Mac OS X driver, see “Create a Data Source” on page 33.

Temporary Table

The Temporary Table feature adds support for creating temporary tables and inserting literal values into temporary tables. Temporary tables are only accessible by the ODBC connection that created them and they will be dropped upon disconnect.

For temporary table CREATE TABLE and INSERT statement syntax please see “Appendix D: Temporary Table CREATE TABLE and INSERT Statements” on page 58.

Get Tables With Query

Hive Server 2 has a limit on the number of tables in a database when handling the GetTables API call. When the number of tables in a database is above the limit it would either run into stack overflow error or timeout error. The exact limit and error depends on the JVM settings.

To address this issue we implemented a workaround in the driver to avoid using the GetTables API call when connecting to Hive Server 2. This feature can be enabled/disabled via the **Get Tables With Query**(**GetTablesWithQuery**) configuration setting.

Active Directory

Hortonworks Hive ODBC Driver with SQL Connector supports Active Directory Kerberos on Windows. There are two prerequisites for using Active Directory Kerberos on Windows:

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

2. The MIT Kerberos Hadoop realm has been configured to trust the Active Directory realm so that users in the Active Directory realm can access services in the MIT Kerberos Hadoop realm. Please refer to the Hortonworks documentation Setting up One-Way Trust with Active Directory for more details.

Appendix A: Configuring Kerberos Authentication for Windows

Active Directory

Hortonworks Hive ODBC Driver with SQL Connector supports Active Directory Kerberos on Windows. There are two prerequisites for using Active Directory Kerberos on Windows:

1. MIT Kerberos is **not** installed on client Windows machine.
2. The MIT Kerberos Hadoop realm has been configured to trust the Active Directory realm so that users in the Active Directory realm can access services in the MIT Kerberos Hadoop realm. Please refer to the Hortonworks documentation Setting up One-Way Trust with Active Directory for more details.

MIT Kerberos

Note: MIT Kerberos is not required if a one-way trust between Hadoop Kerberos realm and the AD (Active Directory) domain has been established. Please refer to the Hortonworks documentation Setting up One-Way Trust with Active Directory for more details.

Download and install MIT Kerberos for Windows 4.0.1

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

Set up the Kerberos configuration file in the default location

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

Consult the MIT Kerberos documentation for more information on configuration.

Set up the Kerberos configuration file in another location

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

1. Obtain a **krb5.conf** configuration file for your Kerberos setup.
2. Store **krb5.conf** in an accessible directory and make note of the full path name.
3. Click the Windows **Start** menu.

4. Right-click **Computer**.
5. Click **Properties**.
6. Click **Advanced system settings**.
7. Click **Environment Variables**.
8. Click **New** for **System variables**.
9. Enter variable name: **KRB5_CONFIG**.
10. Enter variable value: **<absolute pathname to krb5.conf file>**.
11. Click **OK** to save the new variable.
12. Ensure the variable is listed in the **System variables** list.
13. Click **OK** to close Environment Variables Window.
14. Click **OK** to close System Properties Window.

Set up the Kerberos credential cache file

1. Create a writable directory. For example, `c:\temp`
3. Click the Windows **Start** menu.
4. Right-click **Computer**.
5. Click **Properties**.
6. Click **Advanced system settings**.
7. Click **Environment Variables**.
8. Click **New** for System variables.
9. Enter variable name: `KRB5CCNAME`.
10. Enter variable value: <*writable directory from step 1*>`\krb5cache`. For example, `c:\temp\krb5cache`

    Note: `krb5cache` is a regular file (not a directory) managed by the Kerberos software and should not be created by the user. If you receive a permission error when you first use Kerberos, check to make sure that the `krb5cache` file does not exist as a file or a directory.
11. Click **OK** to save the new variable.
12. Ensure the variable is listed in the System variables list.
13. Click **OK** to close Environment Variables Window.
14. Click **OK** to close System Properties Window.
15. Restart your computer to ensure that MIT Kerberos for Windows uses the new settings.

Obtain a ticket for a Kerberos principal using password

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

1. Click the **Start** button.

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

Obtain a ticket for a Kerberos principal using a keytab file

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

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

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

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

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

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

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

Appendix B: Driver Authentication Configuration for Windows

Windows applications that connect with Hive Server 1 or Hive Server 2 will connect in one of two ways:

1. ODBC Data Source
2. Direct driver connection

Applications that connect using ODBC Data Sources will work with Hive Server 2 by sending the appropriate authentication credentials defined in the Data Source.

Applications that connect using a direct driver connection that is Hive Server 1 aware but not Hive Server 2 aware will not have a facility for sending authentication credentials to Hive Server 2. However, the Hortonworks Hive ODBC driver can be configured with authentication credentials using the supplied configuration tool.

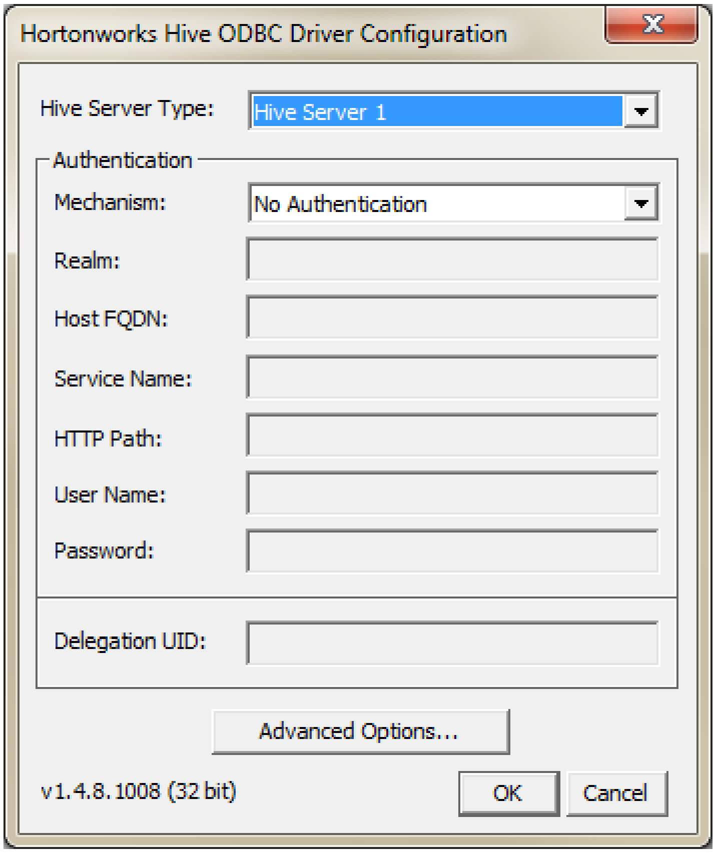
Note: The credentials configured using the configuration application will apply for all connections made using a direct driver connection unless the application is Hive Server 2 aware and requests credentials from the user.

Configure driver authentication

1. Click the **Start** button.
2. Click **All Programs.**
3. Click the **Hortonworks Hive ODBC Driver 1.4 (64-bit)** or the **Hortonworks Hive ODBC Driver 1.4 (32-bit)** program group. If you installed both versions of the driver, you will see two program groups.

   Because drivers are bit-specific, select the version that matches the bitness of your application. For example, a 32-bit driver will only be accessible from 32-bit applications.
4. Click **Driver Configuration** and click **OK** when prompted for administrator permission make modification to the computer.

   Note: You must have administrator access to the computer in order to run this application because it makes changes to the registry.
5. Follow the procedure for configuring the Advanced Options on page 11 and the section “Configuring Authentication in DSN” on page 16 to complete the Hortonworks Hive ODBC Driver Configuration dialog.
6. In the Hortonworks Hive ODBC Driver Configuration dialog, click OK



Hortonworks Hive ODBC Driver Configuration

Hive Server Type: Hive Server 1

Authentication

Mechanism: No Authentication

Realm:

Host FQDN:

Service Name:

HTTP Path:

User Name:

Password:

Delegation UID:

v1.4.8.1008 (32 bit)

Appendix C: Driver Configuration Options

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

Note: You can set configuration options in your odbc.ini and .hortonworks.hiveodbc.ini files. Configuration options set in a .hortonworks.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 .hortonworks.hiveodbc.ini

Key	Default Value	Description
Driver		The name of the installed driver (Hortonworks Hive ODBC Driver) or the absolute path of the Hortonworks Hive ODBC Driver shared object file. (Required)
HOST		The IP address or hostname of the Hive server. (Required)
PORT	10000 for non-HDInsight clusters, 10001 for Windows Azure HDInsight Emulator, and 443 for Windows Azure HDInsight Service	The listening port for the service. (Required)
Schema	default	The name of the database schema to use when a schema is not explicitly specified in a query **Note:** Queries on other schemas can still be issued by explicitly specifying the schema in the query. To determine the appropriate database schema to use, type **show databases** command at the Hive command prompt to inspect your databases. (Optional)

Key	Default Value	Description
DefaultStringColumnLength	255	The maximum data length for string columns. **Note:** Hive does not provide the maximum data length for String columns in the columns metadata. This option allows you to tune the maximum data length for String columns. (Optional)
BinaryColumnLength	32767	The maximum data length for binary columns. **Note:** Hive does not provide the maximum data length for Binary columns in the columns metadata. The option allows you to tune the maximum data length for Binary columns. (Optional)
UseNativeQuery	0	Enabling the UseNativeQuery option using a value of 1 disables the SQL Connector feature. The SQL Connector feature has been added to the driver to apply transformations to the queries emitted by an application to convert them into an equivalent form in HiveQL. If the application is Hive aware and already emits HiveQL, then turning off the SQL Connector feature avoids the extra overhead of query transformation. (Optional)
FastSQLPrepare	0	To enable the FastSQLPrepare option, use a value of 1. Enabling FastSQLPrepare defers query execution to SQLExecute. When using Native Query mode, the driver will execute the HiveQL query to retrieve the result set metadata for SQLPrepare. As a result, SQLPrepare might be slow. If the result set metadata is not required after calling SQLPrepare, then enable FastSQLPrepare. (Optional)

Key	Default Value	Description
EnableAsyncExec	0	Enable/disable the use of asynchronous query execution. **Note:** This option only take effect when connecting to Hive cluster running Hive 0.12.0 or higher. **Note:** Due to the problem in Hive 0.12 reported in JIRA HIVE-5230, Hive returns generic error messages for errors that occur during query execution. To find out about what the actual error message you may turn off asynchronous query execution and execute the query again. Set to 1 to enable. Set to 0 to disable. (Optional)
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)
DecimalColumnScale	10	The maximum number of digits to the right of the decimal point for numeric data types. (Optional)
SSP_		To set a server-side property, use the following syntax where *SSPKey* is the name of the server-side property to set and *SSPValue* is the value to assign to the server-side property: SSP_*SSPKey*=*SSPValue* For example: SSP_mapred.queue.names=myQueue After the driver applies the server-side property, the SSP_ prefix is removed from the DSN entry leaving an entry of *SSPKey*=*SSPValue* (Optional)

Key	Default Value	Description
ApplySSPWithQueries	1	When set to the default value of 1—enabled—each server side property you set is applied by executing a **set *SSPKey=SSPValue*** query when opening a session to the Hive server. Applying server-side properties using queries involves an additional network round trip per server side property when establishing a session to the Hive server. Some Hive Server 2 builds are not compatible with the more efficient method for setting server-side properties that the driver uses when ApplySSPWithQueries is disabled by setting the key value to 0. **Note:** When connecting to a Hive Server 1, ApplySSPWithQueries is always enabled. (Optional)
LCaseSspKeyName	1	Control whether the driver will convert server side property key name to all lower case characters. Set to 1 to enable. Set to 0 to disable. (Optional)
HiveServerType	1	The Hive Server Type. Set it to 1 for Hive Server and 2 for Hive Server 2. (Optional)
AuthMech	0	The authentication mechanism to use. Set the value to 0 for no authentication, 1 for Kerberos, 2 for User Name, 3 for User Name and Password, 4 User Name and Password (SSL), 5 for Windows Azure HDInsight Emulator, 6 for Windows Azure HDInsight Service, 7 for HTTP, or 8 for HTTPS. (Optional)
KrbHostFQDN		The fully qualified domain name of the Hive Server 2 host used. (Required if AuthMech is Kerberos)
KrbServiceName		The Kerberos service principal name of the Hive Server 2. (Required if AuthMech is Kerberos)

Key	Default Value	Description
KrbRealm	Depends on Kerberos configuration.	If there is no default realm configured or the realm of the Hive Server 2 host is different from the default realm for your Kerberos setup, then define the realm of the Hive Server 2 host using this option. (Optional)
HTTPPath		The partial URL corresponding to the Hive server on HDInsight, HTTP, or HTTPS authentication mechanisms. (Optional)
UID		The user name of an existing user on the host running Hive Server 2. **Important:** You must set the hive.server2.authentication property in the hive-site.xml file for the Hive Server 2 to NONE OR The user name set up for Windows Azure HDInsight Service OR The user name set up for Hive Server 2 when using the User Name and Password authentication, User Name and Password (SSL), and HTTPS mechanism. (Required if AuthMech is User Name and Password, User Name and Password (SSL), Windows Azure HDInsight Service, HTTPS)
PWD		The password set up for Windows Azure HDInsight Service, User Name and Password, User Name and Password (SSL), and HTTPS authentication mechanisms. (Required if AuthMech is User Name and Password, User Name and Password (SSL), Windows Azure HDInsight Service, HTTPS)

Key	Default Value	Description
CAIssuedCertNamesMismatch	0	Control whether to allow CA issued SSL certificate name not to match the host name of the Hive server. **Note:** This setting is only applicable to **User Name and Password (SSL)** and **HTTPS** authentication mechanism and will be ignored by other authentication mechanisms. (Optional)
TrustedCerts	**Windows:** For 32 bit driver: C:\Program Files (x86)\Hortonworks Hive ODBC Driver\lib\cacerts.pem For 64 bit driver: C:\Program Files\Hortonworks Hive ODBC Driver\lib\cacerts.pem **Linux:** For 32 bit driver: /usr/lib/hive/lib/native/Linux-i386-32/cacerts.pem For 64 bit driver: /usr/lib/hive/lib/native/Linux-amd64-64/cacerts.pem **Mac OS X:** /usr/lib/hive/lib/native/universal/cacerts.pem	Used to specify the location of the file containing trusted CA certificates for authenticating the Hive server when using SSL. **Note:** This setting is only applicable to **User Name and Password (SSL), Windows Azure HDInsight Service, HTTP,** and **HTTPS** authentication mechanisms, and will be ignored by other authentication mechanisms. **Note:** If this setting is not set then the driver will default to using the trusted CA certificates file installed by the driver. (Optional)

Key	Default Value	Description
AsyncExecPollInterval	100	The time in millisecond between each poll for the status of an asynchronous query execution. **Note:** Asynchronous execution here doesn't mean ODBC asynchronous operations are supported; it only means the RPC call used to execute a query against Hive is asynchronous. (Optional)
ForceSynchronousExec	0	Used to force the driver to do synchronous query execution when connected to HDInsight cluster. Set to 1 to enable. Set to 0 to disable. **Note:** This configuration is only applicable to HDInsight clusters and will be ignored when connecting to non-HDInsight clusters. (Optional)
DelegationUID		Used to delegate all operation against Hive to a user that is different than the authenticated user for the connection. **Note:** This setting is only applicable when connecting to a Hive Server 2 that supports this feature. Otherwise this setting will not take any effect. (Optional)
DriverConfigTakePrecedence	0	Allow driver wide configurations take precedence over connection and DSN settings.
GetTablesWithQuery	0	Control whether to retrieve the names of tables in a database using the GET TABLES query instead of the GetTables Thrift API call. **Note:** This setting is only applicable when connecting to Hive Server 2. Set to 1 to enable. Set to 0 to disable. (Optional)
WebHDFSHost	The Hive Server host.	The hostname or IP address of the machine hosting both the namenode of your Hadoop cluster and the WebHDFS service. (Optional)
WebHDFSPort	50070	The WebHDFS port for the namenode. (Optional)
HDFSUser	hdfs	The name of the HDFS user that the

Key	Default Value	Description
		driver will use to create the necessary files for supporting the Temporary Table feature. (Optional)
HDFSTempTableDir	/tmp/simba	The HDFS directory that the driver will use to store the necessary files for supporting the Temporary Table feature. **Note**: Due to a bug in Hive (see https://issues.apache.org/jira/browse/HIVE-4554) space characters in HDFS path will not work with versions of Hive prior to 0.12.0. (Optional)
TempTableTTL	10	The number of minute a temporary table is guaranteed to exist in Hive after it is created. (Optional)
ADUserNameCase	Unchanged	Control whether the driver should change the user name part of an AD Kerberos UPN to all upper case, all lower case, or remain unchanged. Set to **Upper** to change the user name to all upper case. Set to **Lower** to change the user name to all lower case. Set to **Unchanged** leave the user name unmodified. **Note:** This only used when using Active Directory Kerberos from a Windows client machine to authenticate. (Optional)
ShowHiveSystemTable	0	Controls whether the driver returns SQL_WVARCHAR or SQL_VARCHAR for STRING and VARCHAR columns and SQL_WCHAR or SQL_CHAR for CHAR columns. Set to 0 to return SQL_VARCHAR/SQL_CHAR Set to 1 to return SQL_WVARCHAR/SQL_WCHAR

Key	Default Value	Description
UseUnicodeSqlCharacterTypes	0	Controls whether the driver returns the HIVE_SYSTEM table for catalog function calls such as SQLTables and SQLColumns. Set to 0 to not return the HIVE_SYSTEM table Set to 1 to return the HIVE_SYSTEM table

Table 1 Driver Configuration Options

Appendix D: Temporary Table CREATE TABLE and INSERT Statements

Temporary Table CREATE TABLE Statement

The following DDL syntax for creating temporary table is supported:

<create table statement> := CREATE TABLE <temporary table name> <left paren><column definition list><right paren>

<column definition list> := <column definition>[, <column definition>]*

<column definition> := <column name> <data type>

<temporary table name> := <double quote><number sign><table name><double quote>

<left paren> := (

<right paren> :=)

<double quote> := "

<number sign> := #

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

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

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

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

Temporary Table INSERT Statement

The following is the supported INSERT syntax for temporary table:

<insert statement> := INSERT INTO <temporary table name> <left paren><column name list><right paren> VALUES <left paren><literal value list><right paren>

<column name list> := <column name>[, <column name>]*

<literal value list> := <literal value>[, <literal value>]*

<temporary table name> := <double quote><number sign><table name><double quote>

<left paren> := (

<right paren> :=)

<double quote> := "

<number sign> := #

The following is an example of temporary table INSERT statement:

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

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

Note: INSERT statement is only supported for temporary table.

Third Party Trademarks

Cyrus SASL

Copyright (c) 1998-2003 Carnegie Mellon University. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name "Carnegie Mellon University" must not be used to endorse or promote products derived from this software without prior written permission. For permission or any other legal details, please contact

Office of Technology Transfer

Carnegie Mellon University

5000 Forbes Avenue

Pittsburgh, PA 15213-3890

(412) 268-4387, fax: (412) 268-7395

tech-transfer@andrew.cmu.edu

4. Redistributions of any form whatsoever must retain the following acknowledgment:

"This product includes software developed by Computing Services at Carnegie Mellon University (http://www.cmu.edu/computing/)."

CARNEGIE MELLON UNIVERSITY DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL CARNEGIE MELLON UNIVERSITY BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

ICU License - ICU 1.8.1 and later

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2010 International Business Machines Corporation and others. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.

OpenSSL

Copyright (c) 1998-2008 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgment:
"This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)"

4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.

5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.

6. Redistributions of any form whatsoever must retain the following acknowledgment:
"This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache Hive

Copyright 2008-2011 The Apache Software Foundation.

Apache Thrift

Copyright 2006-2010 The Apache Software Foundation.

Expat

"Copyright (c) 1998, 1999, 2000 Thai Open Source Software Center Ltd

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

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

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