Requirements for MapR

Contents

- Location Connection
  - Hive ODBC Connection
    - SSL Options
- MapR Client
  - MapR Client Configuration
  - Verifying MapR Client Installation
- Client Configuration Files
- Hive External Table
  - ODBC Connection
  - Channel Configuration
- Connecting to MapR

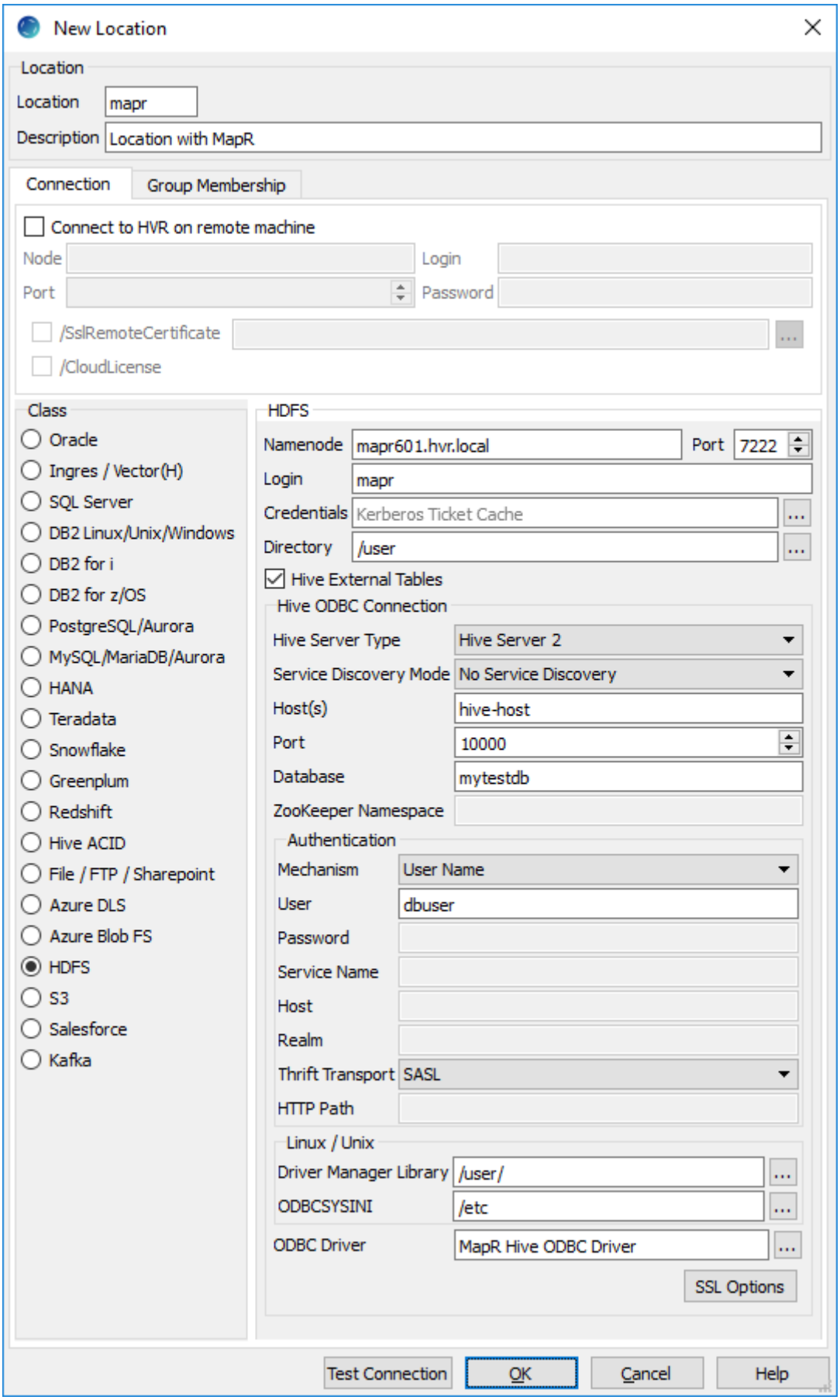
HDFS

Capture	Hub	Integrate
✗	✗	✓

This section describes the requirements, access privileges, and other features of HVR when using MapR for replication. HVR supports the WebHDFS API for reading and writing files from and to MapR.

Location Connection

This section lists and describes the connection details required for creating MapR location in HVR.



Field	Description
Database Connection	

Namenode	The hostname of the MapR Container Location Database (CLDB). **Example:** mapr601.hvr.local
Port	The port on which the MapR CLDB (**Namenode**) is expecting connections. The default port is **7222**. **Example:** 7222
Login	The username to connect HVR to the MapR CLDB (**Namenode**). The username can be either the MapR user or if impersonation is used then the username is the *remotelistener OS user*. **Example:** hvruser
Credentials	The credential (Kerberos Ticket Cache file) for the **Login** to connect HVR to the MapR CLDB (**Namenode**). This field should be left blank to use a **keytab** file for authentication or if Kerberos is not used on the MapR cluster. For more details, see HDFS Authentication and Kerberos.
Directory	The directory path in the MapR CLDB (**Namenode**) to be used for replication. **Example:** /user
Hive External Tables	Enable/Disable Hive ODBC connection configuration for creating Hive external tables above HDFS.

Hive ODBC Connection

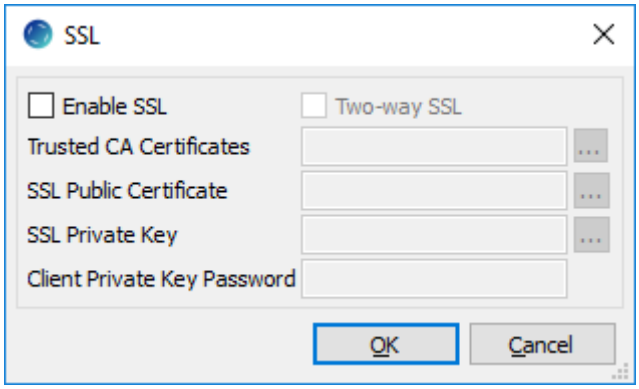
HVR allows you to create Hive External Tables above HDFS which are only used during compare. You can enable/disable the Hive configuration for HDFS in location creation screen using the **Hive External Tables** field.

Field	Description
Hive ODBC Connection	
Hive Server Type	The type of Hive server. Available options: • **Hive Server 1** (default): The driver connects to a Hive Server 1 instance. • **Hive Server 2**: The driver connects to a Hive Server 2 instance.
Service Discovery Mode	The mode for connecting to Hive. This field is enabled only if **Hive Server Type** is **Hive Server 2**. Available options: • **No Service Discovery** (default): The driver connects to Hive server without using the ZooKeeper service. • **ZooKeeper**: The driver discovers Hive Server 2 services using the ZooKeeper service.
Host(s)	The hostname or IP address of the Hive server. When Service Discovery Mode is ZooKeeper, specify the list of ZooKeeper servers in following format **[ZK_Host1]:[ZK_Port1],[ZK_Host2]:[ZK_Port2]**, where **[ZK_Host]** is the IP address or hostname of the ZooKeeper server and **[ZK_Port]** is the TCP port that the ZooKeeper server uses to listen for client connections. **Example:** hive-host
Port	The TCP port that the Hive server uses to listen for client connections. This field is enabled only if **Service Discovery Mode** is **No Service Discovery**. **Example:** 10000

Database	The name of the database schema to use when a schema is not explicitly specified in a query. **Example:** mytestdb
ZooKeeper Namespace	The namespace on ZooKeeper under which Hive Server 2 nodes are added. This field is enabled only if **Service Discovery Mode** is **ZooKeeper**.
Authentication	
Mechanism	The authentication mode for connecting HVR to **Hive Server 2**. This field is enabled only if **Hive Server Type** is **Hive Server 2**. Available options: • **No Authentication** (default) • **User Name** • **User Name and Password** • **Kerberos** • **Windows Azure HDInsight Service** **Since** v5.5.0/2
User	The username to connect HVR to Hive server. This field is enabled only if **Mechanism** is **User Name** or **User Name and Password**. **Example:** dbuser
Password	The password of the **User** to connect HVR to Hive server. This field is enabled only if **Mechanism** is **User Name and Password**.
Service Name	The Kerberos service principal name of the Hive server. This field is enabled only if **Mechanism** is **Kerberos**.
Host	The Fully Qualified Domain Name (FQDN) of the Hive Server 2 host. The value of **Host** can be set as **_HOST** to use the Hive server hostname as the domain name for Kerberos authentication. If **Service Discovery Mode** is disabled, then the driver uses the value specified in the Host connection attribute. If **Service Discovery Mode** is enabled, then the driver uses the **Hive Server 2** host name returned by ZooKeeper. This field is enabled only if **Mechanism** is **Kerberos**.
Realm	The realm of the Hive Server 2 host. It is not required to specify any value in this field if the realm of the Hive Server 2 host is defined as the default realm in Kerberos configuration. This field is enabled only if **Mechanism** is **Kerberos**.
Thrift Transport **Since** v5.5.0/2	The transport protocol to use in the Thrift layer. This field is enabled only if **Hive Server Type** is **Hive Server 2**. Available options: • **Binary** (This option is available only if **Mechanism** is **No Authentication** or **User Name and Password**.) • **SASL** (This option is available only if **Mechanism** is **User Name** or **User Name and Password** or **Kerberos**.) • **HTTP** (This option is not available if **Mechanism** is **User Name**.) For information about determining which Thrift transport protocols your Hive server supports, refer to HiveServer2 Overview and Setting Up HiveServer2 sections in Hive documentation.
HTTP Path **Since** v5.5.0/2	The partial URL corresponding to the Hive server. This field is enabled only if **Thrift Transport** is **HTTP**.
Linux / Unix	

Driver Manager Library	The optional directory path where the ODBC Driver Manager Library is installed. This field is applicable only for Linux/Unix operating system. For a default installation, the ODBC Driver Manager Library is available at **/usr /lib64** and does not need to be specified. However, when UnixODBC is installed in for example **/opt/unixodbc** the value for this field would be **/opt/unixodbc/lib**.
ODBCSYSINI	The optional directory path where **odbc.ini** and **odbcinst.ini** files are located. This field is applicable only for Linux/Unix operating system. For a default installation, these files are available at **/etc** and do not need to be specified. However, when UnixODBC is installed in for example **/opt/unixodbc** th e value for this field would be **/opt/unixodbc/etc**.
ODBC Driver	The user defined (installed) ODBC driver to connect HVR to the Hive server.
SSL Options	Show **SSL Options**.

SSL Options



Field	Description
Enable SSL	Enable/disable (one way) SSL. If enabled, HVR authenticates the Hive server by validating the SSL certificate shared by the Hive server.
Two-way SSL	Enable/disable two way SSL. If enabled, both HVR and Hive server authenticate each other by validating each others SSL certificate. This field is enabled only if **E nable SSL** is selected.
Trusted CA Certificates	The directory path where the **.pem** file containing the server's public SSL certificate signed by a trusted CA is located. This field is enabled only if **Enable SSL** is selected.
SSL Public Certificate	The directory path where the **.pem** file containing the client's SSL public certificate is located. This field is enabled only if **Two-way SSL** is selected.
SSL Private Key	The directory path where the **.pem** file containing the client's SSL private key is located. This field is enabled only if **Two-way SSL** is selected.
Client Private Key Password	The password of the private key file that is specified in **SSL Private Key**. This field is enabled only if **Two-way SSL** is selected.

MapR Client

MapR locations can only be accessed through HVR running on Linux or Windows, and it is not required to run HVR installed on the **Namenode** although it is possible to do so. The MapR client should be present on the server from which HVR will access the MapR (**Namenode**). For more information about installing MapR client, refer to MapR Documentation.

MapR Client Configuration

The following are required on the server from which HVR connects to MapR:

- Install MapR Client
- Install Java Development Kit (JDK), version 1.7 or later
- Install Java Runtime Environment (JRE), version 7 or later
- Set the environment variable **$JAVA_HOME** to the Java installation directory. Ensure that this is the directory that has a bin folder, e.g. if the Java bin directory is d:\java\bin, **$JAVA_HOME** should point to d:\java.
- Set the environment variable **$MAPR_HOME** to the MapR installation directory, or the **hadoop** command line client should be available in the path.

  Since the binary distribution available in Hadoop website lacks Windows-specific executables, a warning about unable to locate **winutils.exe** is displayed. This warning can be ignored for using Hadoop library for client operations to connect to a HDFS server using HVR. However, the performance on integrate location would be poor due to this warning, so it is recommended to use a Windows-specific Hadoop distribution to avoid this warning. For more information about this warning, refer to Hadoop issue HADOOP-10051.

Verifying MapR Client Installation

To verify the MapR client installation,

1. The **MAPR_HOME/bin** directory in MapR installation location should contain the MapR executables in it.
2. Execute the following commands to verify MapR client installation:

   ```
   $JAVA_HOME/bin/java -version
   $MAPR_HOME/bin/hadoop version
   $MAPR_HOME/bin/hadoop classpath
   ```

3. If the MapR client installation is verified successfully then execute the following command to verify the connectivity between HVR and MapR:

   ```
   $MAPR_HOME/bin/hadoop fs -ls hdfs://cluster/
   ```

Client Configuration Files

Client configuration files are not required for HVR to perform replication, however, they can be useful for debugging. Client configuration files contain settings for different services like HDFS, and others. If the HVR integrate server is not part of the cluster, it is recommended to download the configuration files for the cluster so that the MapR client knows how to connect to HDFS.

Hive External Table

HVR allows you to create Hive external tables above HDFS files which are only used during compare. The Hive ODBC connection can be enabled for MapR in the location creation screen by selecting the **Hive External Tables** field (see section Location Connection).

ODBC Connection

HVR uses ODBC connection to the MapR cluster for which it requires the MapR ODBC driver for Hive installed on the server (or in the same network). For more information about using ODBC to connect to HiveServer 2, refer to MapR Documentation.

Channel Configuration

For the file formats (CSV and AVRO) the following action definitions are required to handle certain limitations of the Hive deserialization implementation during Bulk or Row-wise **Compare**:

- For CSV,

Group	Table	Action
HDFS	*	**FileFormat /NullRepresentation=\\N**
HDFS	*	**TableProperties /CharacterMapping="\x00>\\0;\n>\\n;\r>\\r;">\""**
HDFS	*	**TableProperties /MapBinary=BASE64**

- For Avro,

Group	Table	Action
HDFS	*	**FileFormat /AvroVersion=v1_8**

v1_8 is the default value for **FileFormat /AvroVersion**, so it is not mandatory to define this action.

The JSON file format is not supported in MapR.

Connecting to MapR

HVR can connect to MapR with or without using the MapR user impersonation. The configuration/setup requirements differ on each scenarios mentioned below.

- Without MapR User Impersonation
- With MapR User Impersonation

Without MapR User Impersonation

- When hub connects to MapR server using HVR remotelistener installed on the MapR server.
  1. Log in as MapR user and start the remotelistener on the MapR server.
- When hub connects directly to the MapR server or if the integrate server (this is separate from MapR server) connects to the MapR server.
  1. Create a MapR user on the hub/integrate server. Login as **root** user and execute the following commands.

```
groupadd -g2000 mapr

useradd -gmapr -m -u2000 mapr

passwd mapr
```

  2. Login as MapR user and start the remotelistener on hub/integrate server.

With MapR User Impersonation

For more information about MapR user impersonation, refer to MapR Documentation.

- When the hub connects to the MapR server using the remotelistener on the MapR server, the MapR user impersonation is not required.
- When the hub connects directly to the MapR server or if the integrate server (this is separate from MapR server) connects to the MapR server.

1. Login as **root** user on hub/integrate server and modify the **core-site.xml** file available in **/opt/mapr/hadoop/hadoop-2.7.0/etc/hadoop/** directory as shown below:

```
<property>

        <name>hadoop.proxyuser.mapr.hosts</name>

        <value>*</value>

</property>

<property>

        <name>hadoop.proxyuser.mapr.groups</name>

        <value>*</value>

</property>

<property>

    <name>fs.mapr.server.resolve.user</name>

    <value>true</value>

</property>
```

   For more information, refer to MapR Documentation.

2. Create a file in **/opt/mapr/conf/proxy/** that has name of the mapr superuser or any other user. This file can also be copied as shown below,

```
sudo cp /opt/mapr/conf/proxy/mapr /opt/mapr/conf/proxy/hvrremotelistener_os_user
```

3. Export MapR impersonation,

```
export MAPR_IMPERSONATION_ENABLED=true
```

4. Start hvrremotelistener as *hvrremotelistener_os_user*
5. On the MapR server, execute the following using the **config** command,

```
maprcli config save -values {cldb.security.resolve.user:1};
```

   For more information, refer to MapR Documentation.

   a. To verify this update, execute:

```
maprcli config load  -keys cldb.security.resolve.user
```

6. Restart the MapR CLDB.