

<div></div>	<div>POSTED BY</div> <div>ARJUN_KR</div>	<div>POSTED ON</div> <div>DECEMBER 15, 2017</div>	<div>POSTED UNDER</div> <div>DRILL</div>	<div>COMMENTS</div> <div>LEAVE A COMMENT</div>
-------------	--	---	--	--

CONNECT DRILL ODBC USING KERBEROS AUTHENTICATION

Environment:

Drill : 1.10  
 Centos :6.8  
 MapR ODBC Driver : 1.3.8

Purpose:

To setup MapR Drill ODBC connection using Kerberos authentication in Linux environment.

Steps:

- Follow instructions for installing and configuring Drill ODBC driver in linux platform.  
  
<https://drill.apache.org/docs/installing-the-driver-on-linux/>
<https://drill.apache.org/docs/configuring-odbc-on-linux/>
- Identify kerberos principal defined for Drill from drill-override.conf.

```
[arjun@arjun-156 ~]$ cat /opt/mapr/drill/drill-1.10.0/conf/drill-override.conf drill.exec: { .. security: { user.auth.enabled:true, auth.mechanisms:["KERBEROS"], auth.principal:"mapr/maprdrill@MAPR-LAB.LOCAL", auth.keytab:"/opt/mapr/drill/drill_service.keytab" } .. } [arjun@arjun-156 ~]$
```

- Define the ODBC data sources in the `~/odbc.ini` configuration file and configure kerberos authentication as given below.

- Set `AuthenticationType` to Kerberos
- Set `KrbServiceName` and `KrbServiceHost` according to Drill kerberos principal defined in `drill-override.conf`

Drill requires service principal to be of form `primary/instance@REALM`. In this case, set **KrbServiceName** to **primary** part of principal and **KrbServiceHost** to **instance** part of principal.

- Validate the connectivity using DSN defined. Below is sample DSN definition in `odbc.ini`.

```
[MapR Drill 64-bit]
# This key is not necessary and is only to give a description of the data source.
Description=MapR Drill ODBC Driver (64-bit) DSN
# Driver: The location where the ODBC driver is installed to.
Driver=/opt/mapr/drill/lib/64/libdrillodbc_sb64.so
# The DriverUnicodeEncoding setting is only used for SimbaDM
# When set to 1, SimbaDM runs in UTF-16 mode.
# When set to 2, SimbaDM runs in UTF-8 mode.
#DriverUnicodeEncoding=2
# Values for ConnectionType, AdvancedProperties, Catalog, Schema should be set here.
# If ConnectionType is Direct, include Host and Port. If ConnectionType is ZooKeeper, include ZKQuorum and ZKClusterID
# They can also be specified on the connection string.
# AuthenticationType: No authentication; Plain; Kerberos; MapRSASL;
ConnectionType=Direct
HOST=10.10.72.156
PORT=31010
AuthenticationType=Kerberos
UID=
krbSpnConfigurationsRequired=1
KrbServiceName=mapr
KrbServiceHost=maprdrill
AdvancedProperties=CastAnyToVarchar=true;HandshakeTimeout=5;QueryTimeout=180;TimestampTZDisplayTimezone=utc;ExcludedSchemas=sys,INFORMATION_SCHEMA;NumberOfPrefetchBuffers=5;
Catalog=DRILL
Schema=dfs.tmp
```

Below is sample connectivity test using `iodbc` test program.

```
[arjun@arjun-156 ~]$ klist[arjun@arjun-156 ~]$ klistTicket cache: FILE:/tmp/krb5cc_500Default principal: arjun/arjun-156.mapr-lab.local@MAPR-LAB.LOCAL
Valid starting Expires Service principal12/10/17 00:15:41 12/11/17 00:15:41 krbtgt/MAPR-LAB.LOCAL@MAPR-LAB.LOCAL renew until 12/10/17 00:15:41[arjun@arjun-156 ~]$ iodbc test "DSN=MapR Drill 64-bit"iODBC Demonstration programThis program shows an interactive SQL processorDriver Manager: 03.52.0709.0909Driver: 1.3.8.1030 (MapR Drill ODBC Driver)
```

```
SQL>select version from sys.version
version
-----
1.10.0

result set 1 returned 1 rows.

SQL>
```

Notes:

The connection will fail incase of kerberos ticket not available as given below.

```
[arjun@arjun-156 ~]$ iodbctest "DSN=MapR Drill 64-bit"
iODBC Demonstration program
This program shows an interactive SQL processor
Driver Manager: 03.52.0709.0909
1: SQLDriverConnect = [MapR][Drill] (30) User authentication failed. Server message: Authentication failed. Check connection parameters? (30) SQLSTATE=28000
1: ODBC_Connect = [MapR][Drill] (30) User authentication failed. Server message: Authentication failed. Check connection parameters? (30) SQLSTATE=28000

Have a nice day.[arjun@arjun-156 ~]$
[arjun@arjun-156 ~]$
[arjun@arjun-156 ~]$ klist
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_500)
[arjun@arjun-156 ~]$
```

Even with valid ticket available, it may fail if incorrect service principals are provided in DSN. You can look at kerberos logs to identify the issue in such case.

You may get same error with invalid kerberos principal specified in DSN configuration. Below is error log from /var/log/krb5kdc.log with invalid principal given.

```
Dec 14 22:26:49 arjun-156 krb5kdc[22555](info): TGS_REQ (4 etypes {18 17 16 23}) 10.10.72.156: UNKNOWN_SERVER: authtime 0, arjun/arjun-156.mapr-lab.local@MAPR-LAB.LOCAL for
mapr/junk@MAPR-LAB.LOCAL, Server not found in Kerberos database
```

Advertisements

[Report this ad](#)

[Report this ad](#)

	POSTED BY	POSTED ON	POSTED	COMMENTS
	ARJUN_KR	AUGUST 19, 2017	UNDER	LEAVE A
			DRILL	COMMENT

## CONFIGURING DRILL JDBC DRIVER WITH SPLUNK DB CONNECT

This post explains on how to setup Apache Drill JDBC driver with Splunk DB Connect.

### Environment :

- Centos 6.6
- Splunk 6.6.2
- Splunk db connect 3.1.0
- Drill 1.10
- MapR 5.2.0

1) Download the splunk enterprise version for Linux. You have to register account for downloading trial version.  
[https://www.splunk.com/en\\_us/download/splunk-enterprise.htm](https://www.splunk.com/en_us/download/splunk-enterprise.htm)

The RPM downloaded in my case is “splunk-6.6.2-4b804538c686-linux-2.6-x86\_64.rpm”

2) Copy the downloaded rpm to server and install it.

```
# Assuming rpm is copied to /tmp directory in server. Installing to /opt/splunk directory.
rpm -i --prefix=/opt/splunk /tmp/splunk-6.6.2-4b804538c686-linux-2.6-x86_64.rpm
```

3) Download splunk db connect app ‘splunk-db-connect\_310.tgz’ (3.1.0 version) from below URL <https://splunkbase.splunk.com/app/2686/#/overview>.

4) Untar the splunk-db-connect\_310.tgz and copy it to /opt/splunk/splunk/etc/apps

```
tar -xvzf /tmp/splunk-db-connect_310.tgz -C /opt/splunk/splunk/etc/apps
```

5) Start splunk service using below command for first time by accepting license.

```
/opt/splunk/splunk/bin/splunk start --accept-license
```

6) Download MapR Drill JDBC driver from following link – . The DrillJDBC41.zip contains driver jar ‘DrillJDBC41.jar’ and other dependent jars.

http://package.mapr.com/tools/MapR-JDBC/MapR\_Drill/MapRDrill\_jdbc\_v1.5.3.1006/

7) Copy driver jar that contains Driver class (‘DrillJDBC41.jar’) to /opt/splunk/splunk/etc/apps/splunk\_app\_db\_connect/drivers. Create a directory ‘DrillJDBC41-libs’ inside ‘drivers’ and copy dependent jars to this directory.

```
[root@arjun-lab-73 drivers]# ls -ltrh /opt/splunk/splunk/etc/apps/splunk_app_db_connect/drivers

-rw-r--r--. 1 root root 582K Aug 9 01:37 DrillJDBC41.jar
drwxrwxrwx. 2 root root 4.0K Aug 9 01:37 DrillJDBC41-libs
[root@arjun-lab-73 drivers]#
```

8) Define connection type details in below configuration file. /opt/splunk/splunk/etc/apps/splunk\_app\_db\_connect/local/db\_connection\_types.conf.

```
[root@arjun-lab-73 local]# cat db_connection_types.conf
[default]

[drill]
displayName = drill
serviceClass = com.splunk.dbx2.DefaultDBX2JDBC
jdbcDriverClass = com.mapr.drill.jdbc41.Driver
jdbcUrlFormat = jdbc:drill:drillbit=<host>:<port>
useConnectionPool = false
```

9) Connect to Splunk UI and create identities and Drill connection with connection URL specific to environment.

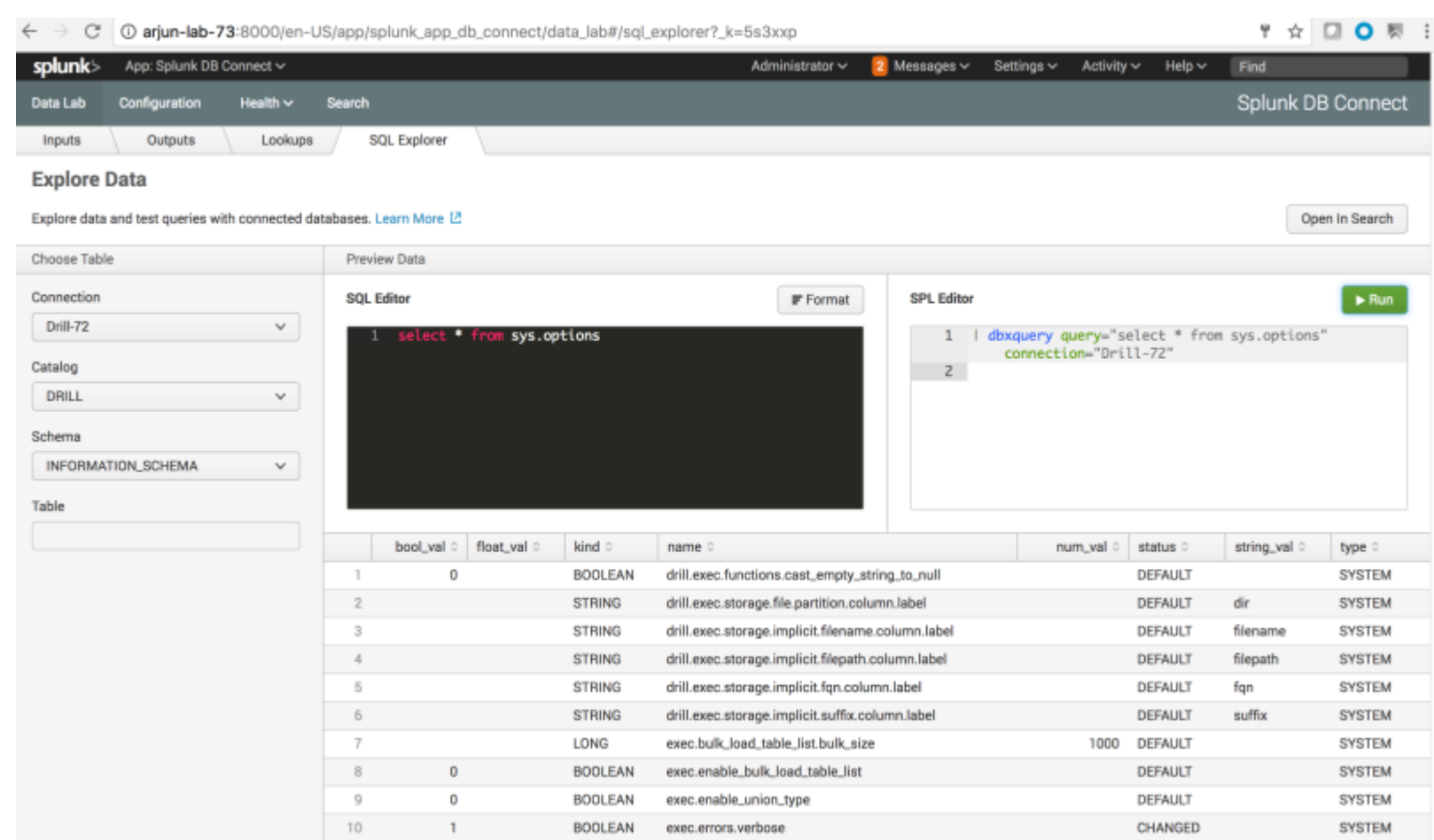
The screenshot shows the 'New Identity' configuration page in the Splunk DB Connect web interface. The page has a top navigation bar with 'Data Lab', 'Configuration', 'Health', and 'Search' tabs. Below the navigation bar, there are tabs for 'Settings' and 'Permissions'. The 'Settings' tab is active, showing fields for 'Identity Name', 'Username', 'Password' (with an 'Optional' checkbox), and 'Windows Authentication Domain' (with a checkbox to 'Use Windows Authentication Domain'). A note at the bottom of the 'Settings' tab states: 'Domain to use with Identity. This field is only effective when using the 'MS-SQL Server Using MS Generic Driver With Windows Authentication' connection type. [Learn More](#)'. At the bottom of the page, there are links for 'About', 'Support', 'File a Bug', 'Documentation', and 'Privacy Policy', along with a copyright notice: '© 2005-2017 Splunk Inc. All rights reserved.'

Create Identity with user name and password for Drill user

The screenshot shows the 'New Connection' configuration page in the Splunk DB Connect web interface. The page has a top navigation bar with 'Data Lab', 'Configuration', 'Health', and 'Search' tabs. Below the navigation bar, there are tabs for 'Settings' and 'Permissions'. The 'Settings' tab is active, showing fields for 'Connection Name', 'Identity' (a dropdown menu with 'mapr' selected), 'Connection Type' (a dropdown menu with 'drill' selected), 'Timezone' (a dropdown menu with 'America/Los\_Angeles : -07:00' selected), and 'JDBC URL Settings'. The 'JDBC URL Settings' section includes fields for 'Host' (arjun-lab-72), 'Port' (31010), and 'Default Database' (dfs.root). A note at the bottom of the 'JDBC URL Settings' section states: 'The usage and meaning of this parameter varies between database vendors. [Learn More](#)'. There is also a checkbox for 'Enable SSL' with a note: 'This is a DB driver flag and may not be supported by all'. On the right side of the page, there is a 'JDBC URL Preview' section showing the generated URL: 'jdbc:drill:drillbit=arjun-lab-72:31010'. Below the preview is a checkbox for 'Edit JDBC URL'.

Create Database connection with identity,host and port details

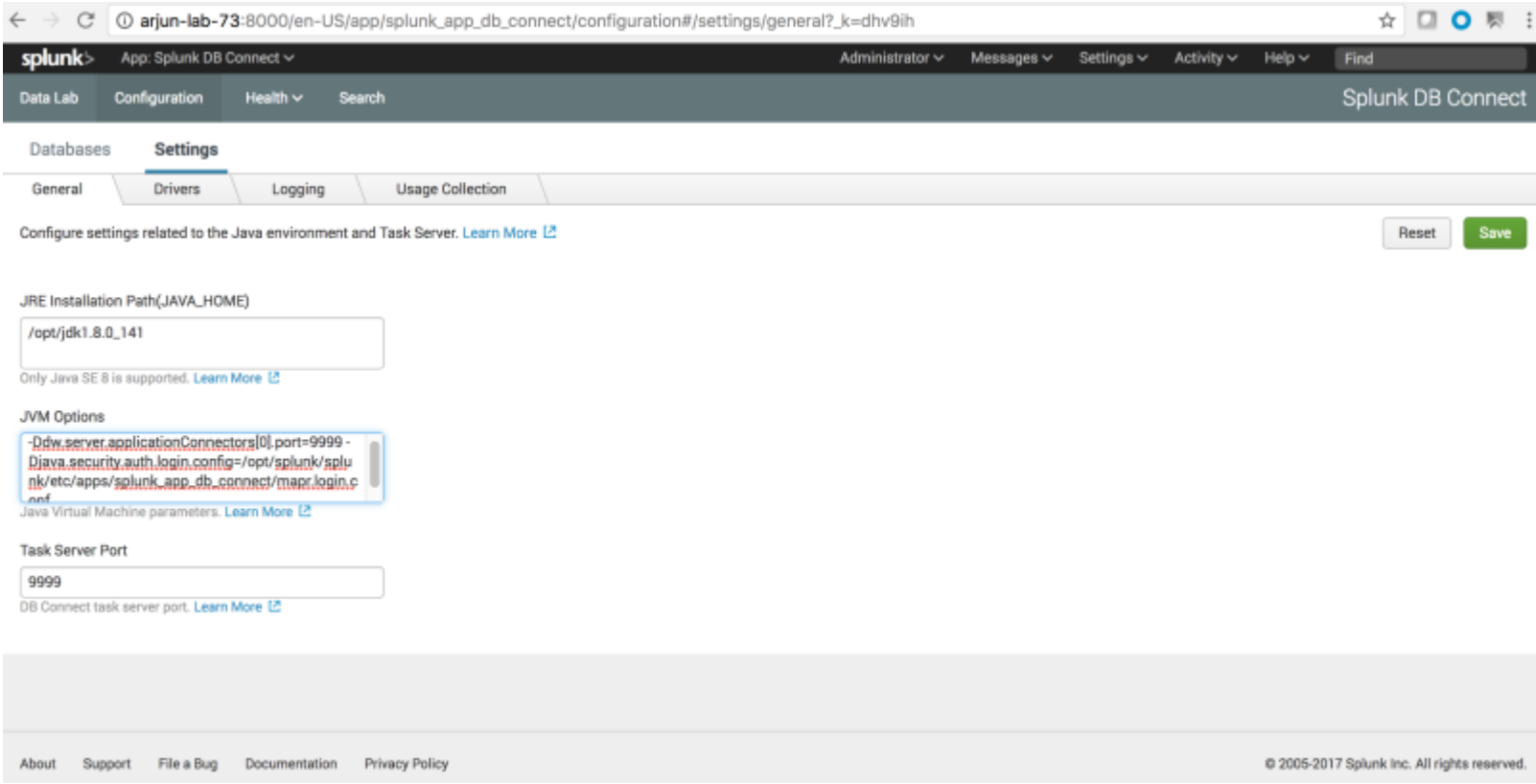
10) Validate the connection with sample queries in SQL explorer.



Test connection using SQL explorer

11) If environment is MapR secure cluster , We would require to set below property as JVM option in DB connect settings.

-Djava.security.auth.login.config=/path/to/mapr.login.conf



Another observation is that – With above settings in MapR secure cluster , We would be able to list databases/Tables from left side panel in SQL explorer. But it failed with below exception while querying from SQL editor window.

java.sql.SQLException: [MapR][Drill]DBCDriver](500150) Error setting/closing connection:

The corresponding job log shows below exception.

```
08-10-2017 11:38:04.664 ERROR ChunkedExternProcessor – stderr: Caused by: com.mapr.support.exceptions.ErrorException: javax.security.sasl.SaslException: Authentication failed unexpectedly. [Caused by java.util.concurrent.ExecutionException: javax.security.sasl.SaslException: Failed to login. [Caused by javax.security.auth.login.LoginException: No LoginModules configured for hadoop_simple]]
```

Below step helped to resolve the issue.

1) Edit /opt/splunk/splunk/etc/apps/splunk\_app\_db\_connect/linux\_x86\_64/bin/command.sh to include ‘-Djava.security.auth.login.config=/path/to/mapr.login.conf’ property. Below is change.

```
[root@arjun-lab-73 bin]# cat /opt/splunk/splunk/etc/apps/splunk_app_db_connect/linux_x86_64/bin/command.sh

SCRIPT=$(readlink -f "$0")
JAVA_PATH_FILE=$(dirname "$SCRIPT")/customized.java.path

if [ -f $JAVA_PATH_FILE ]; then
    JAVA_CMD=`cat $JAVA_PATH_FILE`
elif [ ! -z "$JAVA_HOME" ];then
    JAVA_CMD="$JAVA_HOME/bin/java"
else
    JAVA_CMD="java"
fi

export DRILL_OPTS="-Djava.security.auth.login.config=/opt/splunk/splunk/etc/apps/splunk_app_db_connect/mapr.login.conf"

exec $JAVA_CMD $DRILL_OPTS $@

#exec $JAVA_CMD $@

[root@arjun-lab-73 bin]
```

# EXECUTING CUSTOM PHYSICAL PLAN IN DRILL

**Environment:** Drill

**Purpose:**

To modify and execute physical plan using Drill Web UI.You can test the performance of a physical plan that Drill generates, modify the plan and then re-submit it to Drill. Also this option helps in debugging physical plan issues.

**Steps:**

1) Run EXPLAIN PLAN FOR <query> to see the physical plan for your query.

User-added image

2) Copy the JSON output of the physical plan, and modify as needed.

3) Paste the physical plan into the Query field, select physical plan as query type and click submit. Drill runs the plan and executes the query.

User-added image

You may come across below exception with parquet tables. The issue occurs due to setter less properties in the physical plan for ‘parquet-scan’ operator.

```
Query Failed: An Error Occurred
org.apache.drill.common.exceptions.UserRemoteException:
SYSTEM ERROR: UnsupportedOperationException: Should never call 'set' on setterless property [Error Id: 3c04d66e-f7b9-49c4-a87c-9f30b411af4d on arjun-lab-200-3:31010]
```

To solve this issue, You can remove json element ‘fileSet’ and ‘files’ from “parquet-scan” operator. Below is a sample physical plan with these elements highlighted.

```
{
  "head": {
    "version": 1,
    "generator": {
      "type": "ExplainHandler",
      "info": ""
    },
    "type": "APACHE_DRILL_PHYSICAL",
    "options": [],
    "queue": 0,
    "hasResourcePlan": false,
    "resultMode": "EXEC"
  },
  "graph": [{
    "pop": "parquet-scan",
    "@id": 2,
    "userName": "mapr",
    "entries": [{
      "path": "maprfs:///tmp/emp_parquet"
    }],
    "storage": {
      "type": "file",
      "enabled": true,
      "connection": "maprfs:///",
      "config": null,
      "workspaces": {
        "root": {
          "location": "/",
          "writable": false,
          "defaultInputFormat": null
        },
        "tmp": {
          "location": "/tmp",
          "writable": true,
          "defaultInputFormat": null
        }
      }
    },
    "formats": {
      "psv": {
        "type": "text",
        "extensions": ["tbl"],
        "delimiter": "|"
      },
      "csv": {
        "type": "text",
        "extensions": ["csv"],
        "delimiter": ","
      },
      "tsv": {
        "type": "text",
        "extensions": ["tsv"],
        "delimiter": "\t"
      },
      "parquet": {
        "type": "parquet"
      },
      "json": {
        "type": "json",
        "extensions": ["json"]
      },
      "maprdb": {
        "type": "maprdb"
      }
    }
  ]
}
```

```
        }
    },
    "format": {
        "type": "parquet"
    },
    "columns": ["`*`"],
    "selectionRoot": "maprfs:/tmp/emp_parquet",
    "filter": "true",
    "fileSet": ["/tmp/emp_parquet/0_0_0.parquet"],
    "files": ["/tmp/emp_parquet/0_0_0.parquet"],
    "cost": 14.0
}, {
    "pop": "project",
    "@id": 1,
    "exprs": [{
        "ref": "`*`",
        "expr": "`*`"
    }],
    "child": 2,
    "initialAllocation": 1000000,
    "maxAllocation": 10000000000,
    "cost": 14.0
}, {
    "pop": "screen",
    "@id": 0,
    "child": 1,
    "initialAllocation": 1000000,
    "maxAllocation": 10000000000,
    "cost": 14.0
}]
}
```

You can find settable properties for ‘parquet-scan’ operator in ParquetGroupScan source code.

<https://github.com/apache/drill/blob/1.10.0/exec/java-exec/src/main/java/org/apache/drill/exec/store/parquet/ParquetGroupScan.java>

```
@JsonTypeName("parquet-scan")
public class ParquetGroupScan extends AbstractFileGroupScan {

    ..

    @JsonCreator public ParquetGroupScan( //
        @JsonProperty("userName") String userName,
        @JsonProperty("entries") List<ReadEntryWithPath> entries, //
        @JsonProperty("storage") StoragePluginConfig storageConfig, //
        @JsonProperty("format") FormatPluginConfig formatConfig, //
        @JacksonInject StoragePluginRegistry engineRegistry, //
        @JsonProperty("columns") List<SchemaPath> columns, //
        @JsonProperty("selectionRoot") String selectionRoot, //
        @JsonProperty("cacheFileRoot") String cacheFileRoot, //
        @JsonProperty("filter") LogicalExpression filter
    ) throws IOException, ExecutionSetupException {
        ..
    }
}
```

If this issue occurs for any other operator , you can find corresponding operator ‘pop’ from drillbit exception logs and search in source code repository for ‘@JsonTypeName(“<pop>”)’. The constructor of corresponding class found will have permitted properties for the operator (@JsonProperty values).

	POSTED BY	POSTED ON	POSTED UNDER	COMMENTS
	ARJUN_KR	APRIL 22, 2017	DRILL	LEAVE A COMMENT

## CONNECTING TO DRILL USING ODBC IN LINUX

### Environment:

Drill : 1.8/1.9

Centos :6.8

MapR ODBC Driver : 1.3

### Purpose :

To setup ODBC to connect to Drill using ODBC in Linux .

### Steps :

1. Install iodbc : yum install libiodbc

2. Install MapR ODBC driver for Linux – Follow instructions from <https://drill.apache.org/docs/installing-the-driver-on-linux/>

3. Copy odbc config files to home directory.

```
cd /opt/mapr/drill/Setup/
```

```
cp odbc.ini ~/.odbc.ini
```



```
cp mapr.drillodbc.ini ~/.mapr.drillodbc.ini
cp odbcinstant.ini ~/.odbcinstant.ini
```

4. Update ~/.mapr.drillodbc.ini to point to right ODBCInstLib file. By default it will be ODBCInstLib=libiodbcinstant.so

```
[root@vm75-54 ~]# rpm -qa|grep -i iodbc
libiodbc-3.52.7-1.el6.x86_64
[root@vm75-54 ~]# rpm -ql libiodbc|grep libiodbcinstant
/usr/lib64/libiodbcinstant.so.2
/usr/lib64/libiodbcinstant.so.2.1.19
[root@vm75-54 ~]# cat /home/mapr/.mapr.drillodbc.ini
..
ODBCInstLib=libiodbcinstant.so.2
[root@vm75-54 ~]#
```

4. Set DSN details and enable trace in ~/.odbc.ini

5. Run query using iodbctest “DSN=MapR Drill 64-bit”

```
[mapr@vm75-54 ~]$ iodbctest “DSN=MapR Drill 64-bit”
SQL>select cast(columns[1] as varchar(1024)) from dfs.`tmp`.`employee.csv`
```

— sample .odbc.ini Contents

```
[mapr@vm75-54 ~]$ cat .odbc.ini
[ODBC]
Trace=yes
TraceFile=/tmp/odbc.trace

[ODBC Data Sources]
MapR Drill 32-bit=MapR Drill ODBC Driver 32-bit
MapR Drill 64-bit=MapR Drill ODBC Driver 64-bit

[MapR Drill 32-bit]
# This key is not necessary and is only to give a description of the data source.
Description=MapR Drill ODBC Driver (32-bit) DSN

# Driver: The location where the ODBC driver is installed to.
Driver=/opt/mapr/drill/lib/32/libdrillodbc_sb32.so

# The DriverUnicodeEncoding setting is only used for SimbaDM
# When set to 1, SimbaDM runs in UTF-16 mode.
# When set to 2, SimbaDM runs in UTF-8 mode.
#DriverUnicodeEncoding=2

# Values for ConnectionType, AdvancedProperties, Catalog, Schema should be set here.
# If ConnectionType is Direct, include Host and Port. If ConnectionType is ZooKeeper, include ZKQuorum and ZKClusterID
# They can also be specified on the connection string.
# AuthenticationType: No authentication; Basic Authentication
ConnectionType=Direct
HOST=[HOST]
PORT=[PORT]
ZKQuorum=[Zookeeper Quorum]
ZKClusterID=[Cluster ID]
AuthenticationType=No Authentication
UID=[USERNAME]
PWD=[PASSWORD]
DelegationUID=
AdvancedProperties=CastAnyToVarchar=true;HandshakeTimeout=5;QueryTimeout=180;TimestampTZDisplayTimezone=utc;ExcludedSchemas=sys,INFORMATION_SCHEMA;NumberOfPrefetchBuffers=5;
Catalog=DRILL
Schema=

[MapR Drill 64-bit]
# This key is not necessary and is only to give a description of the data source.
Description=MapR Drill ODBC Driver (64-bit) DSN

# Driver: The location where the ODBC driver is installed to.
Driver=/opt/mapr/drill/lib/64/libdrillodbc_sb64.so

# The DriverUnicodeEncoding setting is only used for SimbaDM
# When set to 1, SimbaDM runs in UTF-16 mode.
# When set to 2, SimbaDM runs in UTF-8 mode.
#DriverUnicodeEncoding=2

# Values for ConnectionType, AdvancedProperties, Catalog, Schema should be set here.
# If ConnectionType is Direct, include Host and Port. If ConnectionType is ZooKeeper, include ZKQuorum and ZKClusterID
# They can also be specified on the connection string.
# AuthenticationType: No authentication; Basic Authentication
ConnectionType=Direct
HOST=XX.XX.XX.XX
PORT=31010
ZKQuorum=
```

ZKClusterID=  
AuthenticationType=No Authentication  
UID=[USERNAME]  
PWD=[PASSWORD]  
DelegationUID=  
AdvancedProperties=CastAnyToVarchar=true;HandshakeTimeout=5;QueryTimeout=180;TimestampTZDisplayTimezone=utc;ExcludedSchemas=sys,INFORMATION\_SCHEMA;NumberOfPrefetchBuffers=5;  
Catalog=DRILL  
Schema=

[mapr@vm75-54 ~]\$


— Sample .mapr.drillodbc.ini

[mapr@vm75-54 ~]\$ cat .mapr.drillodbc.ini  
## – Note that this default DriverManagerEncoding of UTF-32 is for iODBC.  
## – unixODBC uses UTF-16 by default.  
## – If unixODBC was compiled with -DSQL\_WCHART\_CONVERT, then UTF-32 is the correct value.  
##   Execute ‘odbc\_config –cflags’ to determine if you need UTF-32 or UTF-16 on unixODBC  
## – SimbaDM can be used with UTF-8 or UTF-16.  
##   The DriverUnicodeEncoding setting will cause SimbaDM to run in UTF-8 when set to 2 or UTF-16 when set to 1.

[Driver]  
DisableAsync=0  
DriverManagerEncoding=UTF-32  
ErrorMessagesPath=/opt/mapr/drill/ErrorMessage  
LogLevel=0  
LogPath=[LogPath]  
SwapFilePath=/tmp  
ODBCInstLib=libodbcinst.so.2  
[mapr@vm75-54 ~]\$

— Sample odbcinst.ini

[mapr@vm75-54 ~]\$ cat .odbcinst.ini  
[ODBC Drivers]  
MapR Drill ODBC Driver 32-bit=Installed  
MapR Drill ODBC Driver 64-bit=Installed  
  
[MapR Drill ODBC Driver 32-bit]  
Description=MapR Drill ODBC Driver(32-bit)  
Driver=/opt/mapr/drill/lib/32/libdrillodbc\_sb32.so  
  
[MapR Drill ODBC Driver 64-bit]  
Description=MapR Drill ODBC Driver(64-bit)  
Driver=/opt/mapr/drill/lib/64/libdrillodbc\_sb64.so  
[mapr@vm75-54 ~]\$

	POSTED BY	POSTED ON	POSTED UNDER	COMMENTS
	ARJUN_KR	APRIL 22, 2017	DRILL	LEAVE A COMMENT

## ACCESSING MAPR DB JSON TABLE THROUGH APACHE DRILL

### Goal :

To provide read access to non owner users/group to MapR DB json table when queried from Drill.

### Env :

Drill 1.9 and MapR 5.1/5.2 with security and impersonation enabled.

### Symptoms :

Assume user ‘arjun’ creates MapR DB json table ‘/user/arjun/yelp\_user’. When a different user say ‘mapr’ queries the table ,

1. With MapR 5.1 , No rows are returned.
2. With MapR 5.2 , ‘\_id’ column is displayed.

```
1: jdbc:drill:schema=dfs> !connect jdbc:drill:schema=dfs;zk=arjun-156:5181
Enter username for jdbc:drill:schema=dfs;zk=arjun-156:5181: mapr
Enter password for jdbc:drill:schema=dfs;zk=arjun-156:5181: *****
2: jdbc:drill:schema=dfs> select * from dfs.`/user/arjun/yelp_user`
. . . . . > ;
+-----+
|      _id      |
+-----+
| UTS9XcT14H2ZscRif0MYHQ |
+-----+
```



```
1 row selected (0.27 seconds)
2: jdbc:drill:schema=dfs>
```

Root Cause:

By default , only owner user is given read access to MapR DB json tables. The difference in behavior between MapR 5.1 and 5.2 needs to be confirmed. Apparently, it’s same behavior when queried from MapR db shell.

Solution :

With recent MapR DB version , we cannot set permission at file system level using chmod commands. Permissions for MapR tables, column families, and columns are defined by Access Control Expressions (ACEs).

- 1. Identify the column families that need to be given access.

```
[arjun@arjun-156 ~]$ maprccli table cf list -path /user/arjun/yelp_user
readperm  inmemory  cfname  writeperm  traverseperm  compressionperm  memoryperm  compression  ttl          maxversions  minversions
u:arjun   false      default  u:arjun   u:arjun      u:arjun          u:arjun     lz4          2147483647  1           0
[arjun@arjun-156 ~]$
```

- 2. Set read permission to column families identified in previous step. In this case , it’s default. Below command gives read permission to users ‘mapr’ and ‘root’. Make sure to include owner user/group while setting this. If not included , owner may not be able to access the data from mapr db shell/ drill.

```
maprccli table cf edit -path /user/arjun/yelp_user -cfname "default" -readperm "u:mapr|u:root|u:arjun"
```

- 3. Now you can query the table from sqlline as mapr user.
- 4.

```
1: jdbc:drill:schema=dfs> select * from dfs.`/user/arjun/yelp_user`
. . . . . > ;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| _id | average_stars | compliments | fans | friends | name | review_count | type | user_id | votes | yelping_since |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| UTS9XcT14H2ZscRif0MYHQ | 3.59 | {"cool":5.0,"cute":1.0,"funny":8.0,"hot":5.0,"more":2.0,"note":3.0,"photos":1.0,"plain":5.0,"profile":1.0,"writer":3.0} | 7.0 | [
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row selected (0.552 seconds)
1: jdbc:drill:schema=dfs>
```

Please refer to <http://maprdocs.mapr.com/home/SecurityGuide/EnablingTableAuthorizations.html> for more details on MapR DB table authorizations.



POSTED BY

ARJUN\_KR

POSTED ON

APRIL 22, 2017

POSTED UNDER

DRILL

COMMENTS

LEAVE A COMMENT

# DRILL-ON-YARN DEBUGGING OPTIONS

Drill can be run in a YARN-manager cluster like other applications such as Hadoop and Spark. YARN automatically deploys (localizes) the Drill software onto each Drill node and manages the Drill cluster. Drill becomes a long-running application with YARN. You can monitor the Drill-on-YARN cluster using the Application Master web UI.

Drill provides Drill-on-YARN command line tool on Drill-on-YARN client and use it to start, stop, resize, and check the status of the Drill cluster.

```
$DRILL_HOME/bin/drill-on-yarn.sh --site $DRILL_SITE command
```

Below are few options for debugging Drill-on-Yarn application.

- 1. To enable debugging for drillbit containter launched on drill-on-yarn setup.

Set drill.yarn.drillbit.debug-launch = true in \$DRILL\_SITE/drill-on-yarn.conf and restart drill on yarn.

```
drill.yarn: {
  drill-install: {
    ..
  }
  dfs: {
    ..
  }
  drillbit: {
    ..
    debug-launch : "true"
  }
  ...
}
```

Setting this property will display “Drillbit Environment from YARN” properties ,env properties and launch command for drillbit( Invoked by yarn-drillbit.sh). This is useful for finding classpath and other properties used for drillbit. This will be available in stdout logs for drillbit container. Refer to bin/yarn-drillbit.sh for details.

- 2. To enable debug level logging for application master on drill-on-yarn setup.

Set logger level to ‘debug’ in \$DRILL\_SITE/drill-am-log.xml and restart drill on yarn.

```
<logger name="org.apache.drill" additivity="false">
  <level value="debug" />
  <appender-ref ref="STDOUT" />
</logger>

<logger name="org.apache.drill.yarn" additivity="false">
  <level value="debug" />
  <appender-ref ref="STDOUT" />
</logger>
```

3. Debug option for drill-on-yarn.sh script

The drill-on-yarn.sh script is used to invoke drill yarn client by setting DRILL\_CLIENT properties (like DRILL\_CLIENT\_HEAP,DRILL\_CLIENT\_VM\_OPTS).This script provides debug option that can be used to print environment variables and client command for invoking yarn client. If we pass ‘debug’ as first option for script, it will print previously said information (Apparently script usage does not indicate this option).

```
[mapr@arjun-lab-73 bin]$ /opt/mapr/drill/drill-1.9.0/bin/drill-on-yarn.sh
Usage: drill-on-yarn.sh start|stop|status
[mapr@arjun-lab-73 bin]$ /opt/mapr/drill/drill-1.9.0/bin/drill-on-yarn.sh debug
MALLOC_MMAP_MAX_=65536
DRILL_PID_DIR=/opt/mapr/pid
DRILL_MAX_DIRECT_MEMORY=8G
..
CP=/opt/mapr/drill/drill-1.9.0/conf:/opt/mapr/drill/drill-1.9.0/jars/tools/*:/opt/mapr/drill/drill-1.9.0/jars/*:/opt/mapr/drill/drill-1.9.0/jars/ext/*:/opt/mapr/drill/drill-1.9.0/jars/classb/*:/opt/mapr/hadoop/hadoop-2.7.0/etc/hadoop
..
Command: /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.121.x86_64/jre/bin/java -Xms512M -Xmx512M -Dlogback.configurationFile=yarn-client-log.xml -cp /opt/mapr/drill/drill-1.9.0/co
1.9.0/jars/tools/*:/opt/mapr/drill/drill-1.9.0/jars/*:/opt/mapr/drill/drill-1.9.0/jars/ext/*:/opt/mapr/drill/drill-1.9.0/jars/3rdparty/*:/opt/mapr/drill/drill-1.9.0/jars/clas
org.apache.drill.yarn.client.DrillOnYarn debug
[mapr@arjun-lab-73 bin]$
```



Refer to [http://maprdocs.mapr.com/51/Drill/intro\\_drill\\_on\\_yarn.html](http://maprdocs.mapr.com/51/Drill/intro_drill_on_yarn.html) for more details on Drill-on-YARN architecture and configuration.

	POSTED BY	POSTED ON	POSTED	COMMENTS
	ARJUN_KR	APRIL 22, 2017	UNDER	LEAVE A
			DRILL	COMMENT

# CONFIGURING MAPR-SASL AUTHENTICATION WITH DRILL

**Purpose:**  
This post explains on how to configure and connect to Drill with MapR-SASL authentication.

- Prerequisites :**
- Drill 1.10
  - MapR secure cluster
  - MapR JDBC Driver 1.5.1 Version (Available in [https://package.mapr.com/tools/MapR-JDBC/MapR\\_Drill/MapRDrill\\_jdbc\\_v1.5.1.1004/](https://package.mapr.com/tools/MapR-JDBC/MapR_Drill/MapRDrill_jdbc_v1.5.1.1004/))

- Steps:**
1. To enable MapR-SASL authentication for a drillbit, Set the authentication system options for MapR-SASL in drill-override.conf file located in \$DRILL\_HOME/conf.

Set drill.exec.security.auth.mechanisms parameter to include ‘MAPRSASL’ value

```
drill.exec: {
  cluster-id: "my-drillbits",
  zk.connect: "arjun-156:5181",
  security: {
    user.auth.enabled:true,
    auth.mechanisms:["MAPRSASL"]
  }
}
```

2. Generate MapR service ticket for your user and make it readable only to your user.

```
maprlogin generateticket -type service -user arjun -out /home/arjun/maprticket_arjun -duration 3650:0:0
```

3. Set variable ‘MAPR\_TICKETFILE\_LOCATION’ to point to service ticket generated.

```
export MAPR_TICKETFILE_LOCATION=/home/arjun/maprticket_arjun
```