HITACHI
Inspire the Next

Integrating Pentaho with MapR using Apache Drill

HITACHI
Inspire the Next

Change log (if you want to use it):

Date	Version	Author	Changes

Contents

This page intentionally left blank.

Overview

Apache Drill is a schema-free SQL-on-Hadoop tool that lets you run SQL queries against different Hadoop filesystem datasets with various formats such as JSON, CSV, Parquet, HBase, and more. Blending Pentaho Data Integration (PDI) with Apache Drill gives you the flexibility to do data integration work using a SQL interface with MapR.

PDI's support of Drill is limited and provided through our support for JDBC 3/4 drivers. Support of the Apache Drill driver itself is provided through MapR.

Some of the topics covered here include configuring Apache Drill for Pentaho Data Integration, connecting PDI to Drill, and links to recommended settings and best practices.

We assume that you have administrator permissions on the cluster, a MapR Converged Data Platform running with Apache Drill installed, and Apache ZooKeeper running in replicated mode. The examples in this document were created using PDI 7.1.

Our intended audience are data scientists, system administrators, or anyone with a background in PDI or ETL.

The intention of this document is to speak about topics generally; however, these are the specific versions covered here:

Software	Version(s)
Pentaho	6.x, 7.x, 8.0
Apache Drill	1.6 or later
MapR Converged Data Platform	4.x or later

The Components Reference in Pentaho Documentation has a complete list of supported software and hardware.

Before You Begin

Minimum performance and hardware requirements for Pentaho and MapR can be found here:

- Pentaho Components Reference
- MapR Online Documentation

Use Case: SQL Interface to MapR

> *Magnus is an administrator who wants to use PDI to connect via the JDBC interface. He has decided to use Apache Drill's JDBC driver for his MapR SQL interface to make this possible.*

Configure Apache Drill to Work with Pentaho Data Integration

This section guides you through the steps to get to get Drill configured to work with PDI.

You can find details on these topics in the following sections:

- Download and Install Apache Drill JDBC Drivers
- Get the Drill Cluster ID and Construct the URL String
- Configure PDI to Connect to MapR with Drill

The Drill Tutorial pages in MapR's documentation can help you get familiar with Apache Drill.

Step 1: Download and Install Apache Drill JDBC Drivers

Before beginning configuration, make sure that you have downloaded and installed the correct Apache Drill JDBC drivers:

1. Download the latest Drill JDBC Driver package and unzip the file to a location on the same system that has PDI installed.
2. Copy all the files that you just unzipped to this directory in PDI's home directory:

```
..\design-tools\data-integration\lib
```

3. Restart PDI, if it is running.

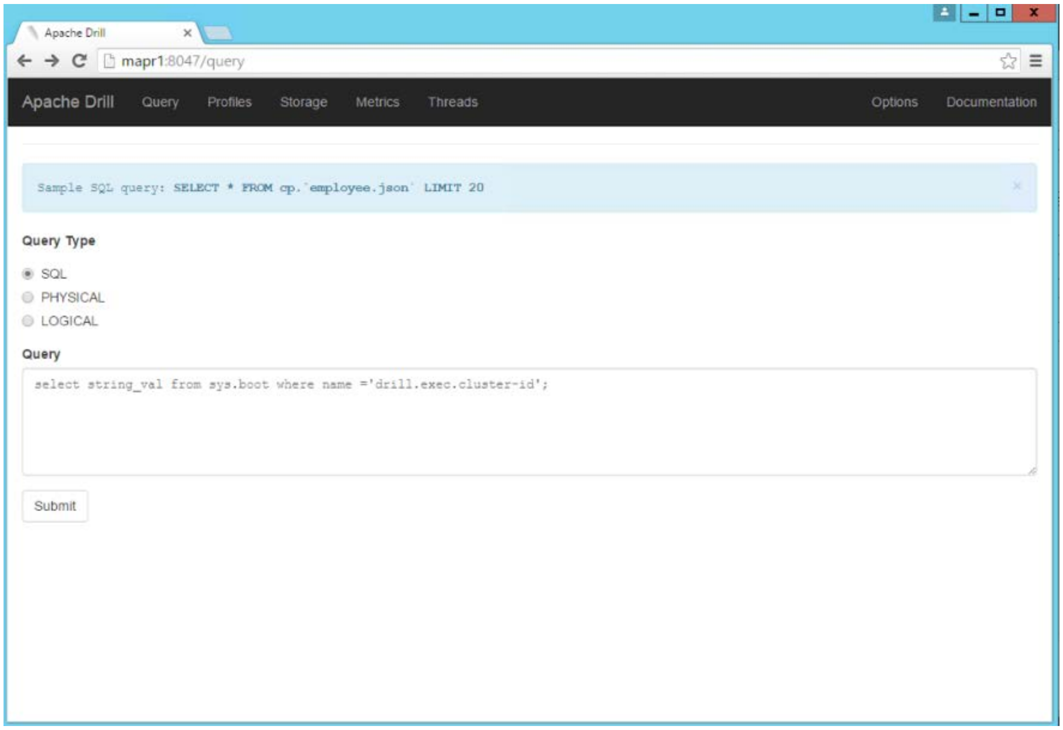
Step 2: Get the Drill Cluster ID and Construct the URL String

After you have the JDBC drivers in place, the next steps consist of getting your Drill cluster ID through ZooKeeper quorum, and then constructing a customer URL string to use for the connection to PDI.

Keep in mind while working through this section that while the default port for ZooKeeper is 2181, ZooKeeper as packaged in MapR *uses 5181 as the default port.*

We are using `mapr520-drillbits` as the cluster ID for this example.

1. To find the Drill cluster ID, first go to the query page on Drill UI: http://mapr1:8047/query



```
Sample SQL query: SELECT * FROM cp.`employee.json` LIMIT 20
```

Query Type

● SQL
○ PHYSICAL
○ LOGICAL

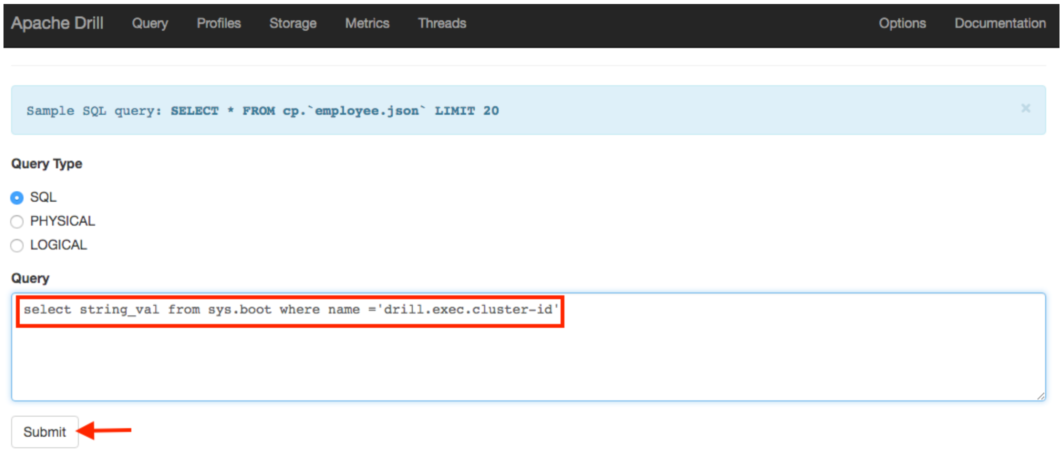
Query

```
select string_val from sys.boot where name ='drill.exec.cluster-id';
```

Figure 1: Query Page on Drill UI

2. Type in this SQL query to get the **Cluster ID** and click **Submit**:

```
select string_val from sys.boot where name ='drill.exec.cluster-id';
```



```
Sample SQL query: SELECT * FROM cp.`employee.json` LIMIT 20
```

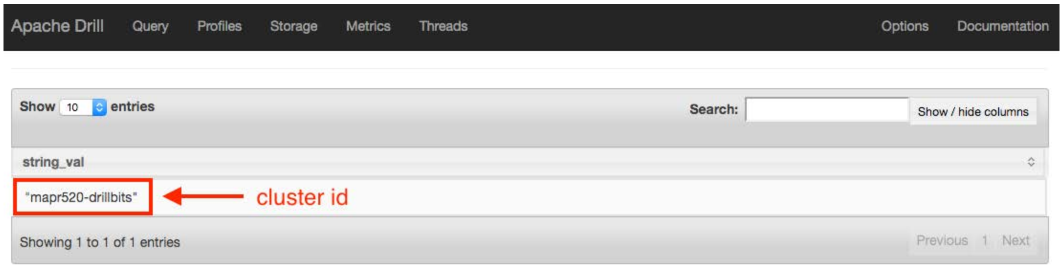
Query Type

● SQL
○ PHYSICAL
○ LOGICAL

Query

```
select string_val from sys.boot where name ='drill.exec.cluster-id'
```

Figure 2: SQL Query for Cluster ID



string_val
"mapr520-drillbits"

Showing 1 to 1 of 1 entries

Figure 3: Cluster ID

3. Now we can construct the **Custom Connection URL** based on the information we obtained above:

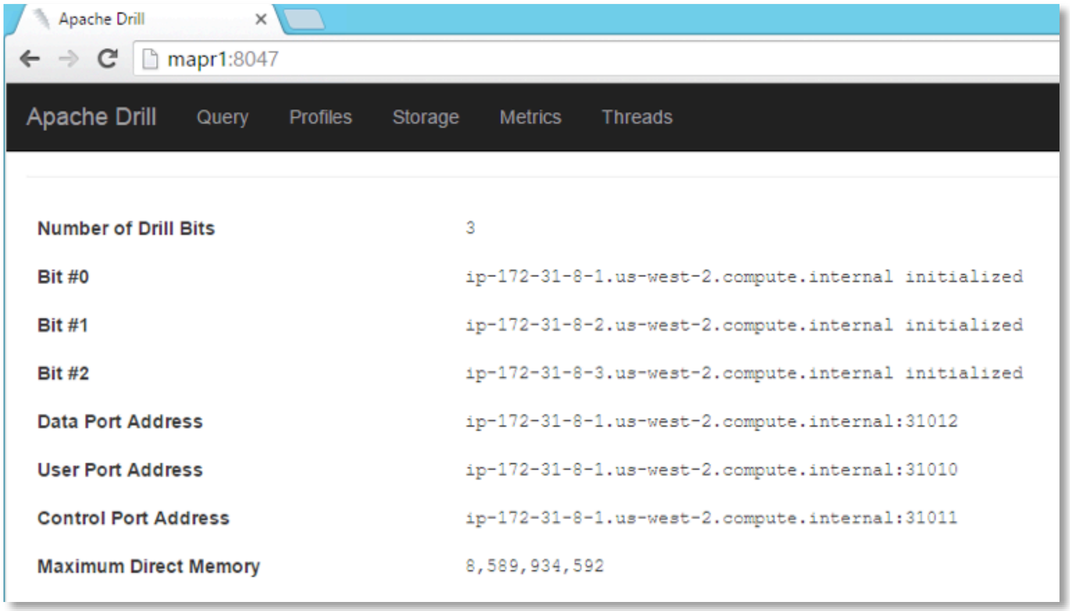
```
jdbc:drill:zk=mapr1:5181,mapr2:5181,mapr3:5181/drill/mapr520-drillbits
```

   This is the URL that you will use later when you are configuring the Drill JDBC connection in PDI.

4. Make sure that the client PDI system can resolve the `hostnames` of all nodes in the Drill cluster, as well as the ZooKeeper quorum, before you begin working with PDI. Do this through DNS configuration or with a Windows system by editing the `hosts` file. For example, your cluster node `hostnames` may be in `C:\Windows\System32\drivers\etc\hosts`. Here is the content of the hosts file, using `ip-172-31-8-* hostnames` as an example:

```
10.0.0.1 mapr1 ip-172-31-8-1.us-west-2.compute.internal
10.0.0.2 mapr2 ip-172-31-8-2.us-west-2.compute.internal
10.0.0.3 mapr3 ip-172-31-8-3.us-west-2.compute.internal
```

5. You can find these `hostnames` through the **Drill UI** of your Drill Bit nodes; for example, http://mapr1:8047/



Number of Drill Bits	3
Bit #0	ip-172-31-8-1.us-west-2.compute.internal initialized
Bit #1	ip-172-31-8-2.us-west-2.compute.internal initialized
Bit #2	ip-172-31-8-3.us-west-2.compute.internal initialized
Data Port Address	ip-172-31-8-1.us-west-2.compute.internal:31012
User Port Address	ip-172-31-8-1.us-west-2.compute.internal:31010
Control Port Address	ip-172-31-8-1.us-west-2.compute.internal:31011
Maximum Direct Memory	8,589,934,592

Figure 4: Drill UI of Drill Bit Nodes

Step 3: Configure PDI to Connect to MapR with Drill

Here are the steps to guide you through the process of configuring PDI to connect to Apache Drill.

1. Open PDI, start a new **Transformation**, then click on the **View** tab in the far left.
2. Expand **Transformation 1**, then right-click on **Database connections** and select **New**.

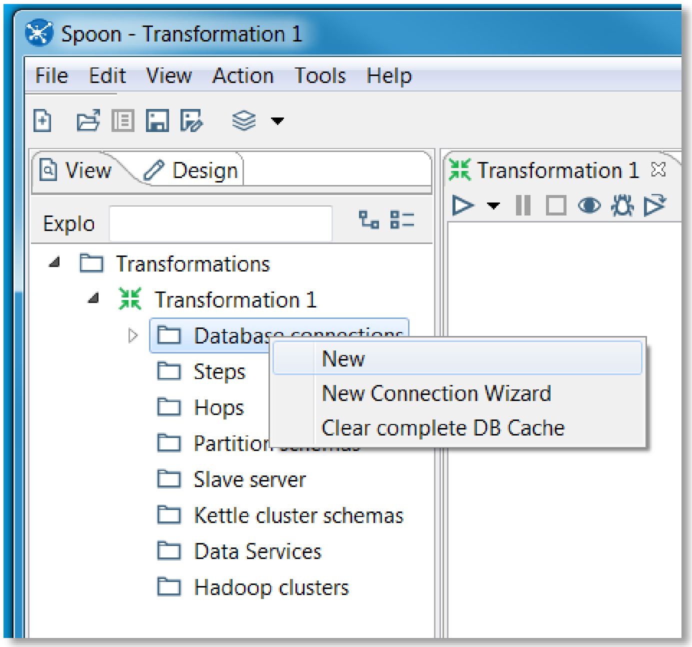
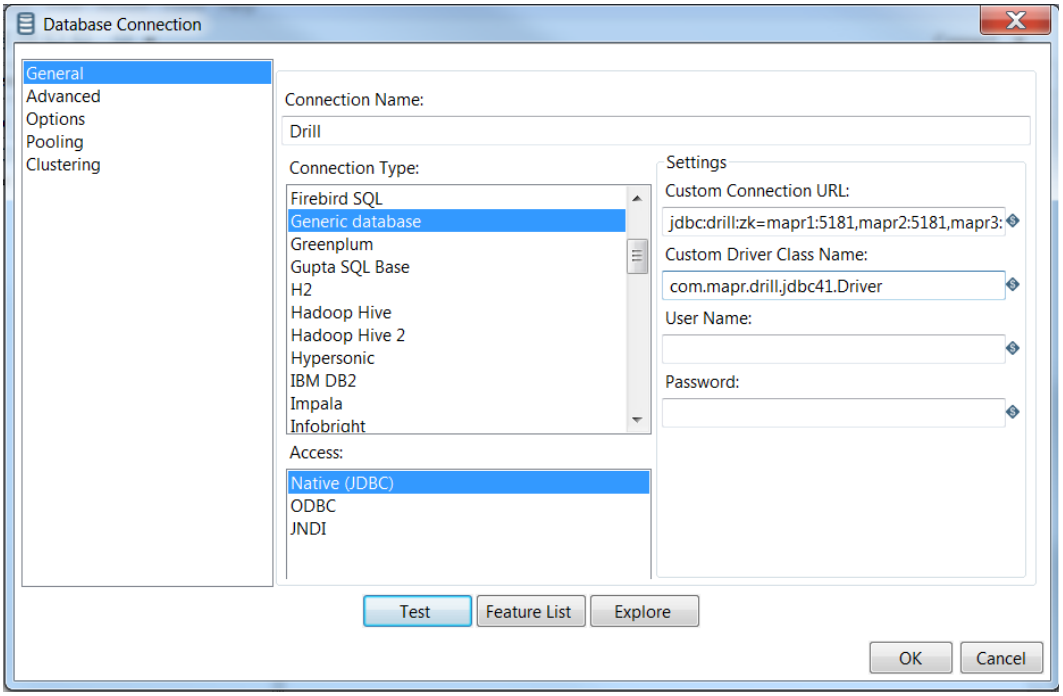


Figure 5: Creating New Database Connection

3. In the Database Connection window:
    a. Name the connection. We are using **Drill** as the **Connection Name** in our example.
    b. Select **Generic Database** for your database type and **Native JDBC** for your access type.
    c. Under **Settings** on the right, copy and paste the **Custom Connection URL** that you created earlier.
    d. Enter the **Custom Driver Class Name**.
    e. Leave the username and password fields empty for now.



Connection Name: Drill

Custom Connection URL: jdbc:drill:zk=mapr1:5181,mapr2:5181,mapr3:

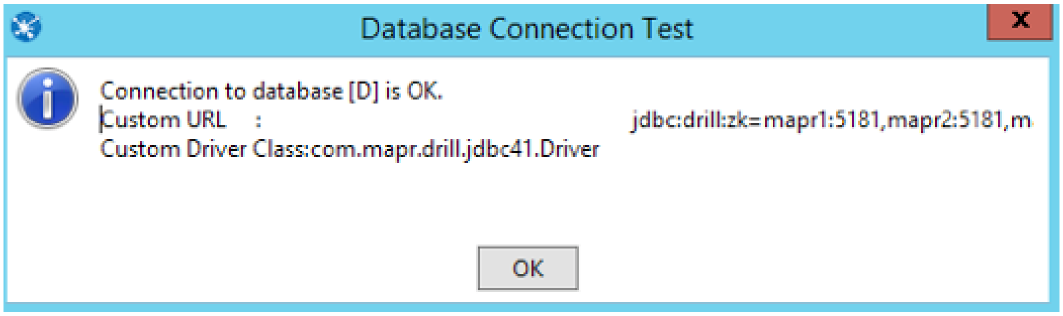
Custom Driver Class Name: com.mapr.drill.jdbc41.Driver

User Name:

Password:

Figure 6: Configuring New Database Connection

4. Click **Test** to verify the connection. You should see a connection success window pop up.



Connection to database [D] is OK.
Custom URL : jdbc:drill:zk=mapr1:5181,mapr2:5181,m
Custom Driver Class:com.mapr.drill.jdbc41.Driver

Figure 7: Testing New Database Configuration

5. If your Database Connection Test does not work:
   a. Verify that the Custom URL string is correct.
   b. Examine your `hosts` file for the PDI client and make sure that it can resolve the `private hostnames` of the MapR cluster.

Settings and Recommended Best Practices

This section provides links to several performance tuning and settings recommendations for Pentaho Data Integration, MapR, and Apache Drill, as well as security and troubleshooting information for Apache Drill.

You can find details on these topics in the following sections:

- Pentaho Data Integration – Recommended Settings and Best Practices
- MapR – Recommended Settings and Best Practices
- Apache Drill

Since MapR tables are prefixed with the back quote character (`), all table names and column names need to be prefixed with the back quote character. Currently there is no setting that will allow this to be done automatically.

Pentaho Data Integration – Recommended Settings and Best Practices

A collection of best practices and recommended settings for Pentaho Data Integration can be found here:

- Pentaho Components Reference
- Pentaho Data Integration
- Pentaho Data Integration Performance Tuning
- Pentaho Data Integration Design Guidelines

MapR – Recommended Settings and Best Practices

In addition to the MapR tutorial and best practices sites, Apache Drill performance tuning and settings recommendations specific to MapR have been collected on the MapR Community website.

- Apache Drill Best Practices from the MapR Drill Team
- MapR: Apache Drill
- MapR Drill Tutorial Site
- MapR Online Documentation

Apache Drill

General Apache Drill information is available here:

- Apache Drill

Security Configuration for Apache Drill

Here are some recommended security configurations for Apache Drill:

- Configuring User Impersonation
- S3 Security Configuration with Drill
- Configuring User Authentication

Troubleshooting Apache Drill

Troubleshooting tips for Apache Drill can be found here:

- Troubleshooting for Apache Drill

Related Information

Here are some links to information that you may find helpful while using this best practices document:

- Apache
    - Apache Drill
    - Apache Drill Best Practices from the MapR Drill Team
    - Configuring User Authentication
    - Configuring User Impersonation
    - S3 Security Configuration with Drill
    - Troubleshooting for Apache Drill
    - Running Replicated ZooKeeper
- MapR
    - Drill JDBC Driver package
    - Drill Tutorial pages
    - MapR: Apache Drill
    - MapR Online Documentation
- Pentaho
    - Components Reference
    - JDBC 3/4 drivers
    - Pentaho Data Integration
    - Pentaho Data Integration Performance Tuning
    - PDI Techniques - Design Guidelines
- Wikipedia
    - Hosts (File)

Finalization Checklist

This checklist is designed to be added to any implemented project that uses this collection of best practices, to verify that all items have been considered and reviews have been performed.

Name of the Project:____

Date of the Review:____

Name of the Reviewer:____

Item	Response	Comments
Did you install the Apache Drill JDBC drivers?	YES_______ NO_______	
Did you get the drill cluster ID and construct the URL string?	YES_______ NO_______	
Did you configure PDI to connect to MapR with Drill?	YES_______ NO_______	