

# Notification

## Table of contents

1 Notification for a New Partition.....	2
2 Notification for a Set of Partitions.....	3
3 Server Configuration.....	4

Since HCatalog 0.2 provides notifications for certain events happening in the system. This way applications such as Oozie can wait for those events and schedule the work that depends on them. The current version of HCatalog supports two kinds of events:

- Notification when a new partition is added
- Notification when a set of partitions is added

No additional work is required to send a notification when a new partition is added: the existing addPartition call will send the notification message.

## 1 Notification for a New Partition

To receive notification that a new partition has been added, you need to follow these three steps.

1. To start receiving messages, create a connection to a message bus as shown here:

```
ConnectionFactory connFac = new ActiveMQConnectionFactory(amqurl);
Connection conn = connFac.createConnection();
conn.start();
```

2. Subscribe to a topic you are interested in. When subscribing on a message bus, you need to subscribe to a particular topic to receive the messages that are being delivered on that topic.

- The topic name corresponding to a particular table is stored in table properties and can be retrieved using the following piece of code:

```
HiveMetaStoreClient msc = new HiveMetaStoreClient(hiveConf);
String topicName = msc.getTable("mydb",
    "myTbl").getParameters().get(HCatConstants.HCAT_MSGBUS_TOPIC_NAME);
```

- Use the topic name to subscribe to a topic as follows:

```
Session session = conn.createSession(true, Session.SESSION_TRANSACTED);
Destination hcatTopic = session.createTopic(topicName);
MessageConsumer consumer = session.createConsumer(hcatTopic);
consumer.setMessageListener(this);
```

3. To start receiving messages you need to implement the JMS interface MessageListener, which, in turn, will make you implement the method onMessage(Message msg). This method will be called whenever a new message arrives on the message bus. The message contains a partition object representing the corresponding partition, which can be retrieved as shown here:

```

@Override
public void onMessage(Message msg) {
    // We are interested in only add_partition events on this table.
    // So, check message type first.

    if(msg.getStringProperty(HCatConstants.HCAT_EVENT).equals(HCatConstants.HCAT_ADD_PARTITION_EVENT))
    {
        Object obj = ((ObjectMessage)msg).getObject();
    }
}

```

You need to have a JMS jar in your classpath to make this work. Additionally, you need to have a JMS provider's jar in your classpath. HCatalog is tested with ActiveMQ as a JMS provider, although any JMS provider can be used. ActiveMQ can be obtained from: <http://activemq.apache.org/activemq-550-release.html> .

## 2 Notification for a Set of Partitions

Sometimes a user wants to wait until a collection of partitions is finished. For example, you may want to start processing after all partitions for a day are done. However, HCatalog has no notion of collections or hierarchies of partitions. To support this, HCatalog allows data writers to signal when they are finished writing a collection of partitions. Data readers may wait for this signal before beginning to read.

The example code below illustrates how to send a notification when a set of partitions has been added.

To signal, a data writer does this:

```

HiveMetaStoreClient msc = new HiveMetaStoreClient(conf);

// Create a map, specifying partition key names and values
Map<String,String> partMap = new HashMap<String, String>();
partMap.put("date", "20110711");
partMap.put("country", "*");

// Mark the partition as "done"
msc.markPartitionForEvent("mydb", "mytbl", partMap, PartitionEventType.LOAD_DONE);

```

To receive this notification, the consumer needs to do the following:

1. Repeat steps one and two from above to establish the connection to the notification system and to subscribe to the topic.
2. Receive the notification as shown in this example:

```

HiveMetaStoreClient msc = new HiveMetaStoreClient(conf);

// Create a map, specifying partition key names and values
Map<String,String> partMap = new HashMap<String, String>();
partMap.put("date", "20110711");

```

```
partMap.put("country", "*");

// Mark the partition as "done"
msc.markPartitionForEvent("mydb", "mytbl", partMap, PartitionEventType.LOAD_DONE);
```

If the consumer has registered with the message bus and is currently live, it will get the callback from the message bus once the producer marks the partition as "done". Alternatively, the consumer can ask explicitly for a particular partition from the metastore. The following code illustrates the usage from a consumer's perspective:

```
// Enquire to metastore whether a particular partition has been marked or not.
boolean marked = msc.isPartitionMarkedForEvent("mydb", "mytbl", partMap,
    PartitionEventType.LOAD_DONE);

// Or register to a message bus and get asynchronous callback.
ConnectionFactory connFac = new ActiveMQConnectionFactory(amqurl);
Connection conn = connFac.createConnection();
conn.start();
Session session = conn.createSession(true, Session.SESSION_TRANSACTED);
Destination hcatTopic = session.createTopic(topic);
MessageConsumer consumer = session.createConsumer(hcatTopic);
consumer.setMessageListener(this);

public void onMessage(Message msg) {

    MapMessage mapMsg = (MapMessage)msg;
    Enumeration<String> keys = mapMsg.getMapNames();

    // Enumerate over all keys. This will print key-value pairs specifying the
    // particular partition 44which was marked done. In this case, it will print:
    // date : 20110711
    // country: *

    while(keys.hasMoreElements()){
        String key = keys.nextElement();
        System.out.println(key + " : " + mapMsg.getString(key));
    }
    System.out.println("Message: "+msg);
```

### 3 Server Configuration

To enable notification, you need to configure the server (see below).

To disable notification, you need to leave `hive.metastore.event.listeners` blank or remove it from `hive-site.xml`.

#### Enable JMS Notifications

You need to make (add/modify) the following changes to the `hive-site.xml` file of your HCatalog server to turn on notifications.

```

<property>
<name>hive.metastore.event.expiry.duration</name>
<value>300L</value>
<description>Duration after which events expire from events table (in seconds)</
description>
</property>

<property>
<name>hive.metastore.event.clean.freq</name>
<value>360L</value>
<description>Frequency at which timer task runs to purge expired events in metastore(in
seconds).</description>
</property>

<property>
<name>msgbus.brokerurl</name>
<value>tcp://localhost:61616</value>
<description></description>
</property>

<property>
<name>msgbus.username</name>
<value></value>
<description></description>
</property>

<property>
<name>msgbus.password</name>
<value></value>
<description></description>
</property>

```

For the server to start with support for notifications, the following must be in the classpath:

- (a) activemq jar
- (b) jndi.properties file with properties suitably configured for notifications

Then, follow these steps:

1. HCatalog server start script is \$YOUR\_HCAT\_SERVER/share/hcatalog/scripts/hcat\_server\_start.sh
2. This script expects classpath to be set by the AUX\_CLASSPATH environment variable.
3. Therefore set AUX\_CLASSPATH to satisfy (a) and (b) above.
4. jndi.properties file is located at \$YOUR\_HCAT\_SERVER/etc/hcatalog/jndi.properties
5. You need to uncomment and set the following properties in this file: -
  - java.naming.factory.initial = org.apache.activemq.jndi.ActiveMQInitialContextFactory
  - java.naming.provider.url = tcp://localhost:61616 (this is activemq url in your setup)

### Topic Names

If tables are created while the server is configured for notifications, a default topic name is automatically set as table property. To use notifications with tables created previously

(previous HCatalog installations or created prior to enabling notifications), you will have to manually set a topic name, an example will be:

```
$YOUR_HCAT_CLIENT_HOME/bin/hcat -e "ALTER TABLE access SET hcat.msgbus.topic.name=  
$TOPIC_NAME"
```

You then need to configure your activemq Consumer(s) to listen for messages on the topic you gave in \$TOPIC\_NAME. A good default policy for TOPIC\_NAME = "\$database.\$table" (that is a literal dot)