Storage Formats

As of version 0.4, HCatalog uses Hive's SerDe class to serialize and deserialize data. SerDes are provided for RCFile, CSV text, JSON text, and SequenceFile formats.

Users can write SerDes for custom formats using the instructions at https://cwiki.apache.org/confluence/display/Hive/SerDe.

Usage from Hive

As of 0.4, Hive and HCatalog share the same storage abstractions, and thus, you can read from and write to HCatalog tables from within Hive, and vice versa.

However, Hive does not know where to find the HCatalog jar by default, so if you use any features that have been introduced by HCatalog, such as a table using the JSON SerDe, you might get a "class not found" exception. In this situation, before you run Hive, set environment variable `HIVE_AUX_JARS_PATH` to the directory with your HCatalog jar. (If the examples in the Installation document were followed, that should be `/usr/local/hcat/share/hcatalog/`).

CTAS Issue with JSON SerDe

Using the Hive CREATE TABLE ... AS SELECT command with a JSON SerDe results in a table that has column headers such as "`_col0`", which can be read by HCatalog or Hive but cannot be easily read by external users. To avoid this issue, create the table in two steps instead of using CTAS:

1. CREATE TABLE ...
2. INSERT OVERWRITE TABLE ... SELECT ...

See HCATALOG-436 for details.