

# Hadoop: Customer-Managed UDAF Installation

You must manually install AtScale UDAFs if you choose "Customer Managed" as the Custom Function Installation Mode during data warehouse configuration. The "Customer Managed" UDAF management mode is required for MAPR data warehouses.

To facilitate the manual installation and upgrade of AtScale UDAF functions on the Hadoop cluster, AtScale 7.4 and up allows administrators to specify the Custom Function Installation Mode when configuring an AtScale data warehouse connection. The default mode is "None." Choosing "Customer Managed" gives the responsibility of installing and upgrading the AtScale UDAFs to the Hadoop administrator.

The following instructions explain how a data warehouse administrator can install or upgrade AtScale UDAFs on a Hadoop cluster.



**Important:** To connect to MAPR, you must complete the following prerequisites and UDAF installation process.

## Hadoop: Before You Begin

- ▲ Set the AtScale Data Warehouse Custom Function Installation Mode to "Customer Managed" when configuring the AtScale data warehouse connection. See [Adding Hadoop Warehouses](#).
- ▲ Determine which SQL engine you have configured, and follow the corresponding procedure below.

## Hadoop: Procedure For Hive And Spark SQL Engines

1. After running the installation, copy the desired `com.atscale.honeybee.honeybee-*` JAR files to a location on your HDFS cluster:

1. Find the honeybee JAR files on your AtScale host by logging on the host as root (or a sudo user) and execute:

```
find ./ -name com.atscale.honeybee.honeybee*
```

The default location depends on your OS and package manager. When using RPM on CentOS, the JAR files are located here by default:

▲ `./pkg/<atscale-version>/lib/com.atscale.honeybee.honeybee-hive-4.0.68-assembly.jar`

2. Copy the desired JAR file to the Hadoop file system location. By default, AtScale stores these functions in the HDFS home directory for the Hadoop `atscale` user. For example:

```
hdfs dfs -put ./pkg/<atscale-version>/lib/com.atscale.honeybee.honeybee-hive-4.0.68-assembly.jar /user/atscaler/atscale/engine/
```

2. If upgrading AtScale, or if you've previously installed the functions, then you must first drop the functions by connecting to your data warehouse with a SQL client and executing the following SQL commands:

```
DROP FUNCTION atscale_honeybee_version;  
DROP FUNCTION hll_aggregate;  
DROP FUNCTION hll_aggregate_estimate;  
DROP FUNCTION hll_aggregate_merge;  
DROP FUNCTION hll_estimate;  
DROP FUNCTION quantile_estimate;  
DROP FUNCTION quantile_sketch;  
DROP FUNCTION quantile_sketch_merge;  
DROP FUNCTION quantilefromsketch;
```

3. Substitute the HDFS JAR location in the following example commands with the location of the JAR file on your Hadoop file system. Register the functions by executing the CREATE FUNCTION command for each function. For example:

```
CREATE FUNCTION atscale_honeybee_version AS 'com.atscale.honeybee.HoneyBeeVersionUDF' USING JAR
'hdfs:///user/atscaler/atscale/engine/com.atscale.honeybee.honeybee-hive-4.0.68-assembly.jar';

CREATE FUNCTION hll_aggregate AS 'com.atscale.honeybee.hyperloglog.HyperLogLogAggregateUDA' USING JAR
'hdfs:///user/atscaler/atscale/engine/com.atscale.honeybee.honeybee-hive-4.0.68-assembly.jar';

CREATE FUNCTION hll_aggregate_estimate AS
'com.atscale.honeybee.hyperloglog.HyperLogLogAggregateEstimateUDA' USING JAR
'hdfs:///user/atscaler/atscale/engine/com.atscale.honeybee.honeybee-hive-4.0.68-assembly.jar';

CREATE FUNCTION hll_aggregate_merge AS 'com.atscale.honeybee.hyperloglog.HyperLogLogAggregateMergeUDA'
USING JAR 'hdfs:///user/atscaler/atscale/engine/com.atscale.honeybee.honeybee-hive-4.0.68-assembly.jar';

CREATE FUNCTION hll_estimate AS 'com.atscale.honeybee.hyperloglog.HyperLogLogEstimateUDA' USING JAR
'hdfs:///user/atscaler/atscale/engine/com.atscale.honeybee.honeybee-hive-4.0.68-assembly.jar';

CREATE FUNCTION quantile_estimate AS 'com.atscale.honeybee.quantile.QuantileEstimateUDA' USING JAR
'hdfs:///user/atscaler/atscale/engine/com.atscale.honeybee.honeybee-hive-4.0.68-assembly.jar';

CREATE FUNCTION quantilefromsketch AS 'com.atscale.honeybee.quantile.QuantileEstimateFromSketchUDF' USING
JAR 'hdfs:///user/atscaler/atscale/engine/com.atscale.honeybee.honeybee-hive-4.0.68-assembly.jar';

CREATE FUNCTION quantile_sketch AS 'com.atscale.honeybee.quantile.QuantileAggregateUDA' USING JAR
'hdfs:///user/atscaler/atscale/engine/com.atscale.honeybee.honeybee-hive-4.0.68-assembly.jar';

CREATE FUNCTION quantile_sketch_merge AS 'com.atscale.honeybee.quantile.QuantileMergeUDA' USING JAR
'hdfs:///user/atscaler/atscale/engine/com.atscale.honeybee.honeybee-hive-4.0.68-assembly.jar';
```

## Hadoop: Procedure For Impala SQL Engines

1. After running the AtScale installation, extract the desired com.atscale.honeybee.honeybee-\* JAR files and copy the extracted shared objects to a location on your HDFS cluster:

1. Find the honeybee JAR files on your AtScale host by logging on the host as root (or a sudo user) and execute:

```
find ./ -name com.atscale.honeybee.honeybee*
```

The default location depends on your OS and package manager. When using RPM on CentOS, the JAR files are located here by default:

▲ `./pkg/<atscale-version>/lib/com.atscale.honeybee.honeybee-impala-1.7.1.52.jar`

2. Extract the shared object `libatscale-uda-1.7.1.so` from `com.atscale.honeybee.honeybee-impala-1.7.1.52.jar`. For example:

```
unzip /opt/atscale/versions/7.4.*/pkg/engine-7.4.*/lib/com.atscale.honeybee.honeybee-impala-1.7.1.52.jar -d /tmp
```

3. Copy the shared object to the Hadoop file system location. By default, AtScale stores these functions in the HDFS home directory for the Hadoop `atscale` user. For example:

```
hdfs dfs -put /tmp/libatscale-uda-1.7.1.so /user/atScaler/atscale/engine/
```

2. If upgrading AtScale, or if you've previously installed the functions, then you must first drop the functions by connecting to your data warehouse with a SQL client and executing the following SQL commands:

```
DROP FUNCTION atscale_honeybee_version();
DROP FUNCTION FUNCTION TO_DATE_AS(TIMESTAMP);
DROP FUNCTION quantileFromSketch(STRING, DOUBLE);

-- hll_aggregate
DROP AGGREGATE FUNCTION hll_aggregate(string);
DROP AGGREGATE FUNCTION hll_aggregate(int);
DROP AGGREGATE FUNCTION hll_aggregate(double);
DROP AGGREGATE FUNCTION hll_aggregate(bigint);
DROP AGGREGATE FUNCTION hll_aggregate(float);

-- hll_aggregate_estimate
DROP AGGREGATE FUNCTION hll_aggregate_estimate(string);

-- hll_aggregate_merge
DROP AGGREGATE FUNCTION hll_aggregate_merge(string);

-- hll_estimate
DROP AGGREGATE FUNCTION hll_estimate(string);
DROP AGGREGATE FUNCTION hll_estimate(int);
DROP AGGREGATE FUNCTION hll_estimate(double);
DROP AGGREGATE FUNCTION hll_estimate(bigint);
DROP AGGREGATE FUNCTION hll_estimate(float);

DROP FUNCTION quantile_sketch(INT, INT);
DROP FUNCTION quantile_sketch(DOUBLE, INT);
DROP FUNCTION quantile_sketch(BIGINT, INT);
DROP FUNCTION quantile_sketch(FLOAT, INT);
DROP FUNCTION quantile_sketch_estimate(STRING, DOUBLE);
DROP FUNCTION quantile_sketch_merge(STRING);

DROP FUNCTION quantile_estimate(INT, DOUBLE, INT);
DROP FUNCTION quantile_estimate(DOUBLE, DOUBLE, INT);
DROP FUNCTION quantile_estimate(BIGINT, DOUBLE, INT);
DROP FUNCTION quantile_estimate(FLOAT, DOUBLE, INT);
```

3. Substitute the HDFS location in the following example commands with the location of the shared object file on your Hadoop file system. Register the functions by executing the CREATE FUNCTION command for each function. For example:

```

CREATE FUNCTION atscale_honeybee_version() RETURNS string LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' SYMBOL='AtScaleHoneyBeeVersion';
CREATE FUNCTION TO_DATE_AS(TIMESTAMP) RETURNS string LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' SYMBOL='ToDate';
CREATE FUNCTION quantileFromSketch(STRING, DOUBLE) RETURNS double LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' SYMBOL='QuantileEstimateFromSketchUDF';

CREATE AGGREGATE FUNCTION hll_aggregate(STRING) RETURNS STRING LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' UPDATE_FN='HyperLogLogAggregateUpdate';
CREATE AGGREGATE FUNCTION hll_aggregate(INT) RETURNS STRING LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' UPDATE_FN='HyperLogLogAggregateUpdate';
CREATE AGGREGATE FUNCTION hll_aggregate(DOUBLE) RETURNS STRING LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' UPDATE_FN='HyperLogLogAggregateUpdate';
CREATE AGGREGATE FUNCTION hll_aggregate(BIGINT) RETURNS STRING LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' UPDATE_FN='HyperLogLogAggregateUpdate';
CREATE AGGREGATE FUNCTION hll_aggregate(FLOAT) RETURNS STRING LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' UPDATE_FN='HyperLogLogAggregateUpdate';
CREATE AGGREGATE FUNCTION hll_aggregate_estimate(STRING) RETURNS STRING LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' UPDATE_FN='HyperLogLogAggregateEstimateUpdate';
CREATE AGGREGATE FUNCTION hll_aggregate_merge(STRING) RETURNS STRING LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' UPDATE_FN='HyperLogLogAggregateMergeUpdate';

CREATE AGGREGATE FUNCTION hll_estimate(STRING) RETURNS STRING LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' UPDATE_FN='HyperLogLogEstimateUpdate';
CREATE AGGREGATE FUNCTION hll_estimate(INT) RETURNS STRING LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' UPDATE_FN='HyperLogLogEstimateUpdate';
CREATE AGGREGATE FUNCTION hll_estimate(DOUBLE) RETURNS STRING LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' UPDATE_FN='HyperLogLogEstimateUpdate';
CREATE AGGREGATE FUNCTION hll_estimate(BIGINT) RETURNS STRING LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' UPDATE_FN='HyperLogLogEstimateUpdate';
CREATE AGGREGATE FUNCTION hll_estimate(FLOAT) RETURNS STRING LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' UPDATE_FN='HyperLogLogEstimateUpdate';

CREATE AGGREGATE FUNCTION quantile_sketch(INT, INT) RETURNS STRING LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' UPDATE_FN='QuantileAggregateUpdate';
CREATE AGGREGATE FUNCTION quantile_sketch(DOUBLE, INT) RETURNS STRING LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' UPDATE_FN='QuantileAggregateUpdate';
CREATE AGGREGATE FUNCTION quantile_sketch(BIGINT, INT) RETURNS STRING LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' UPDATE_FN='QuantileAggregateUpdate';
CREATE AGGREGATE FUNCTION quantile_sketch(FLOAT, INT) RETURNS STRING LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' UPDATE_FN='QuantileAggregateUpdate';
CREATE AGGREGATE FUNCTION quantile_sketch_estimate(STRING, DOUBLE) RETURNS STRING LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' UPDATE_FN='QuantileAggregateEstimateUpdate';
CREATE AGGREGATE FUNCTION quantile_sketch_merge(STRING) RETURNS STRING LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' UPDATE_FN='QuantileAggregateMergeUpdate';

CREATE AGGREGATE FUNCTION quantile_estimate(INT, DOUBLE, INT) RETURNS STRING LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' UPDATE_FN='QuantileEstimateUpdate';
CREATE AGGREGATE FUNCTION quantile_estimate(DOUBLE, DOUBLE, INT) RETURNS STRING LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' UPDATE_FN='QuantileEstimateUpdate';
CREATE AGGREGATE FUNCTION quantile_estimate(BIGINT, DOUBLE, INT) RETURNS STRING LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' UPDATE_FN='QuantileEstimateUpdate';
CREATE AGGREGATE FUNCTION quantile_estimate(FLOAT, DOUBLE, INT) RETURNS STRING LOCATION
'hdfs:///user/atscaler/atscale/engine/libatscale-uda-1.7.1.so' UPDATE_FN='QuantileEstimateUpdate';

```