Configuration Properties

Created by Carl Steinbach, last modified by Zoltan Haindrich on Jul 09, 2018

Hive Configuration Properties

- Hive Configuration Properties
  - Query and DDL Execution
    - SerDes and I/O
      - SerDes
      - I/O
    - File Formats
      - RCFile Format
      - ORC File Format
      - Parquet
    - Vectorization
  - MetaStore
    - Hive Metastore Connection Pooling Configuration
    - Hive Metastore HBase
  - HiveServer2
    - HiveServer2 Web UI
  - Spark
    - Remote Spark Driver
  - Tez
  - LLAP
    - LLAP Client
    - LLAP Web Services
    - LLAP Cache
    - LLAP I/O
    - LLAP CBO
    - LLAP Metrics
    - LLAP UDF Security
    - LLAP Security
  - Transactions and Compactor
    - Transactions
    - Compactor
    - Compaction History
  - Indexing
  - Statistics
    - Runtime Filtering
  - Authentication and Authorization
    - Restricted/Hidden/Internal List and Whitelist
      - Whitelist for SQL Standard Based Hive Authorization
    - Hive Client Security
    - Hive Metastore Security
    - SQL Standard Based Authorization
  - Archiving
  - Locking
  - Metrics
  - Clustering
  - Regions
  - Command Line Interface
  - HBase StorageHandler
  - Hive Web Interface (HWI) (component removed as of Hive 2.2.0)
  - Replication
  - Blobstore (i.e. Amazon S3)
  - Test Properties
- HCatalog Configuration Properties
- WebHCat Configuration Properties

This document describes the Hive user configuration properties (sometimes called *parameters*, *variables*, or *options*), and notes which releases introduced new properties.

The canonical list of configuration properties is managed in the `HiveConf` Java class, so refer to the `HiveConf.java` file for a complete list of configuration properties available in your Hive release.

For information about how to use these configuration properties, see Configuring Hive. That document also describes administrative configuration properties for setting up Hive in the Configuration Variables section. Hive Metastore Administration describes additional configuration properties for the metastore.

> **Version information**
> As of Hive 0.14.0 (HIVE-7211), a configuration name that starts with "hive." is regarded as a Hive system property. With the hive.conf.validation option true (default), any attempts to set a configuration property that starts with "hive." which is not registered to the Hive system will throw an exception.

Query and DDL Execution

hive.execution.engine

- Default Value: `mr` (deprecated in Hive 2.0.0 – see below)
- Added In: Hive 0.13.0 with HIVE-6103 and HIVE-6098

Chooses execution engine. Options are: `mr` (Map Reduce, default), `tez` (Tez execution, for Hadoop 2 only), or `spark` (Spark execution, for Hive 1.1.0 onward).

While `mr` remains the default engine for historical reasons, it is itself a historical engine and is deprecated in the Hive 2 line (HIVE-12300). It may be removed without further warning.

See Hive on Tez and Hive on Spark for more information, and see the Tez section and the Spark section below for their configuration properties.

hive.execution.mode

- Chooses whether query fragments will run in container or in llap
- Default Value: `container`
- `Valid settings`
  - `container: launch containers`
  - `llap: utilize llap nodes during execution of tasks`
- Added In: Hive 2.0 with HIVE-9460

mapred.reduce.tasks

- Default Value: `-1`
- Added In: Hive 0.1.0

The default number of reduce tasks per job. Typically set to a prime close to the number of available hosts. Ignored when mapred.job.tracker is "local". Hadoop set this to 1 by default, whereas Hive uses -1 as its default value. By setting this property to -1, Hive will automatically figure out what should be the number of reducers.

hive.exec.reducers.bytes.per.reducer

- Default Value: `1,000,000,000` prior to Hive 0.14.0; 256 MB (`256,000,000`) in Hive 0.14.0 and later
- Added In: Hive 0.2.0; default changed in 0.14.0 with HIVE-7158 (and HIVE-7917)

Size per reducer. The default in Hive 0.14.0 and earlier is 1 GB, that is, if the input size is 10 GB then 10 reducers will be used. In Hive 0.14.0 and later the default is 256 MB, that is, if the input size is 1 GB then 4 reducers will be used.

hive.exec.reducers.max

- Default Value: `999` prior to Hive 0.14.0; `1009` in Hive 0.14.0 and later
- Added In: Hive 0.2.0; default changed in 0.14.0 with HIVE-7158 (and HIVE-7917)

Maximum number of reducers that will be used. If the one specified in the configuration property **mapred.reduce.tasks** is negative, Hive will use this as the maximum number of reducers when automatically determining the number of reducers.

hive.jar.path

- Default Value: (empty)
- Added In: Hive 0.2.0 or earlier

The location of hive_cli.jar that is used when submitting jobs in a separate jvm.

hive.aux.jars.path

- Default Value: (empty)
- Added In: Hive 0.2.0 or earlier

The location of the plugin jars that contain implementations of user defined functions (UDFs) and SerDes.

hive.reloadable.aux.jars.path

- Default Value: (empty)
- Added In: Hive 0.14.0 with HIVE-7553

The locations of the plugin jars, which can be comma-separated folders or jars. They can be renewed (added, removed, or updated) by executing the Beeline reload command without having to restart HiveServer2. These jars can be used just like the auxiliary classes in **hive.aux.jars.path** for creating UDFs or SerDes.

hive.exec.scratchdir

- Default Value: `/tmp/${user.name}` in Hive 0.2.0 through 0.8.0; `/tmp/hive-${user.name}` in Hive 0.8.1 through 0.14.0; or `/tmp/hive` in Hive 0.14.0 and later
- Added In: Hive 0.2.0; default changed in 0.8.1 and in 0.14.0 with HIVE-6847 and HIVE-8143

Scratch space for Hive jobs. This directory is used by Hive to store the plans for different map/reduce stages for the query as well as to stored the intermediate outputs of these stages.

Hive 0.14.0 and later: HDFS root scratch directory for Hive jobs, which gets created with write all (733) permission. For each connecting user, an HDFS scratch directory ${**hive.exec.scratchdir**}/<username> is created with ${**hive.scratch.dir.permission**}.

Also see **hive.start.cleanup.scratchdir** and **hive.scratchdir.lock**. When running Hive in local mode, see **hive.exec.local.scratchdir**.

hive.scratch.dir.permission

- Default Value: `700`
- Added In: Hive 0.12.0 with HIVE-4487

The permission for the user-specific scratch directories that get created in the root scratch directory. (See **hive.exec.scratchdir**.)

hive.exec.local.scratchdir

- Default Value: `/tmp/${user.name}`
- Added In: Hive 0.10.0 with HIVE-1577

Scratch space for Hive jobs when Hive runs in local mode. Also see **hive.exec.scratchdir**.

hive.hadoop.supports.splittable.combineinputformat

- Default Value: `false`
- Added In: Hive 0.6.0 with HIVE-1280
- Removed In: Hive 2.0.0 with HIVE-11376

Whether to combine small input files so that fewer mappers are spawned.

hive.map.aggr

- Default Value: `true` in Hive 0.3 and later; `false` in Hive 0.2
- Added In: Hive 0.2.0

Whether to use map-side aggregation in Hive Group By queries.

hive.groupby.skewindata

- Default Value: `false`
- Added In: Hive 0.3.0

Whether there is skew in data to optimize group by queries.

hive.groupby.mapaggr.checkinterval

- Default Value: `100000`
- Added In: Hive 0.3.0

Number of rows after which size of the grouping keys/aggregation classes is performed.

hive.new.job.grouping.set.cardinality

- Default Value: `30`
- Added In: Hive 0.11.0 with HIVE-3552

Whether a new map-reduce job should be launched for grouping sets/rollups/cubes.

For a query like "select a, b, c, count(1) from T group by a, b, c with rollup;" four rows are created per row: (a, b, c), (a, b, null), (a, null, null), (null, null, null). This can lead to explosion across the map-reduce boundary if the cardinality of T is very high, and map-side aggregation does not do a very good job.

This parameter decides if Hive should add an additional map-reduce job. If the grouping set cardinality (4 in the example above) is more than this value, a new MR job is added under the assumption that the orginal "group by" will reduce the data size.

hive.mapred.local.mem

- Default Value: `0`
- Added In: Hive 0.3.0

For local mode, memory of the mappers/reducers.

hive.map.aggr.hash.force.flush.memory.threshold

- Default Value: `0.9`
- Added In: Hive 0.7.0 with HIVE-1830

The maximum memory to be used by map-side group aggregation hash table. If the memory usage is higher than this number, force to flush data.

hive.map.aggr.hash.percentmemory

- Default Value: `0.5`
- Added In: Hive 0.2.0

Portion of total memory to be used by map-side group aggregation hash table.

hive.map.aggr.hash.min.reduction

- Default Value: `0.5`
- Added In: Hive 0.4.0

Hash aggregation will be turned off if the ratio between hash table size and input rows is bigger than this number. Set to 1 to make sure hash aggregation is never turned off.

hive.optimize.groupby

- Default Value: `true`
- Added In: Hive 0.5.0

Whether to enable the bucketed group by from bucketed partitions/tables.

hive.optimize.countdistinct

- Default Value: `true`
- Added In: Hive 3.0.0 with HIVE-16654

Whether to rewrite count distinct into 2 stages, i.e., the first stage uses multiple reducers with the count distinct key and the second stage uses a single reducer without key.

hive.optimize.remove.sq_count_check

- Default Value: `false`
- Added In: Hive 3.0.0 with HIVE-16793

Whether to remove an extra join with sq_count_check UDF for scalar subqueries with constant group by keys.

hive.multigroupby.singlemr

- Default Value: `false`
- Added In: Hive 0.8.0 with HIVE-2056
- Removed In: Hive 0.9.0 by HIVE-2621 (see **hive.multigroupby.singlereducer**)

Whether to optimize multi group by query to generate a single M/R job plan. If the multi group by query has common group by keys, it will be optimized to generate a single M/R job. (This configuration property was removed in release 0.9.0.)

hive.multigroupby.singlereducer

- Default Value: `true`
- Added In: Hive 0.9.0 with HIVE-2621

Whether to optimize multi group by query to generate a single M/R job plan. If the multi group by query has common group by keys, it will be optimized to generate a single M/R job.

hive.optimize.cp

- Default Value: `true`
- Added In: Hive 0.4.0 with HIVE-626
- Removed In: Hive 0.13.0 with HIVE-4113

Whether to enable column pruner. (This configuration property was removed in release 0.13.0.)

hive.optimize.index.filter

- Default Value: `false`
- Added In: Hive 0.8.0 with HIVE-1644

Whether to enable automatic use of indexes.

Note: See Indexing for more configuration properties related to Hive indexes.

hive.optimize.ppd

- Default Value: `true`
- Added In: Hive 0.4.0 with HIVE-279, default changed to true in Hive 0.4.0 with HIVE-626

Whether to enable predicate pushdown (PPD).

Note: Turn on **hive.optimize.index.filter** as well to use file format specific indexes with PPD.

hive.optimize.ppd.storage

- Default Value: `true`
- Added In: Hive 0.7.0

Whether to push predicates down into storage handlers. Ignored when hive.optimize.ppd is false.

hive.ppd.remove.duplicatefilters

- Default Value: `true`
- Added In: Hive 0.8.0

During query optimization, filters may be pushed down in the operator tree. If this config is true, only pushed down filters remain in the operator tree, and the original filter is removed. If this config is false, the original filter is also left in the operator tree at the original place.

hive.ppd.recognizetransivity

- Default Value: `true`
- Added In: Hive 0.8.1

Whether to transitively replicate predicate filters over equijoin conditions.

hive.join.emit.interval

- Default Value: `1000`
- Added In: Hive 0.2.0

How many rows in the right-most join operand Hive should buffer before emitting the join result.

hive.join.cache.size

- Default Value: `25000`
- Added In: Hive 0.5.0

How many rows in the joining tables (except the streaming table) should be cached in memory.

hive.mapjoin.bucket.cache.size

- Default Value: `100`
- Added In: Hive 0.5.0 (replaced by **hive.smbjoin.cache.rows** in Hive 0.12.0)

How many values in each key in the map-joined table should be cached in memory.

hive.mapjoin.followby.map.aggr.hash.percentmemory

- Default Value: `0.3`
- Added In: Hive 0.7.0 with HIVE-1830

Portion of total memory to be used by map-side group aggregation hash table, when this group by is followed by map join.

hive.smalltable.filesize *or* hive.mapjoin.smalltable.filesize

- Default Value: `25000000`
- Added In: Hive 0.7.0 with HIVE-1642: **hive.smalltable.filesize** (replaced by **hive.mapjoin.smalltable.filesize** in Hive 0.8.1)
- Added In: Hive 0.8.1 with HIVE-2499: **hive.mapjoin.smalltable.filesize**

The threshold (in bytes) for the input file size of the small tables; if the file size is smaller than this threshold, it will try to convert the common join into map join.

hive.mapjoin.localtask.max.memory.usage

- Default Value: `0.90`
- Added In: Hive 0.7.0 with HIVE-1808 and HIVE-1642

This number means how much memory the local task can take to hold the key/value into an in-memory hash table. If the local task's memory usage is more than this number, the local task will be aborted. It means the data of small table is too large to be held in memory.

hive.mapjoin.followby.gby.localtask.max.memory.usage

- Default Value: `0.55`
- Added In: Hive 0.7.0 with HIVE-1830

This number means how much memory the local task can take to hold the key/value into an in-memory hash table when this map join is followed by a group by. If the local task's memory usage is more than this number, the local task will abort by itself. It means the data of the small table is too large to be held in memory.

hive.mapjoin.check.memory.rows

- Default Value: `100000`
- Added In: Hive 0.7.0 with HIVE-1808 and HIVE-1642

The number means after how many rows processed it needs to check the memory usage.

hive.ignore.mapjoin.hint

- Default Value: `true`
- Added In: Hive 0.11.0 with HIVE-4042

Whether Hive ignores the mapjoin hint.

hive.smbjoin.cache.rows

- Default Value: `10000`
- Added In: Hive 0.12.0 with HIVE-4440 (replaces **hive.mapjoin.bucket.cache.size**)

How many rows with the same key value should be cached in memory per sort-merge-bucket joined table.

hive.mapjoin.optimized.keys

- Default Value: `true`
- Added In: Hive 0.13.0 with HIVE-6429 and HIVE-6188

- Removed In: Hive 1.1.0 with HIVE-9331

Whether a MapJoin hashtable should use optimized (size-wise) keys, allowing the table to take less memory. Depending on the key, memory savings for the entire table can be 5-15% or so.

hive.mapjoin.optimized.hashtable

- Default Value: `true`
- Added In: Hive 0.14.0 with HIVE-6430

Whether Hive should use a memory-optimized hash table for MapJoin. Only works on Tez and Spark, because memory-optimized hash table cannot be serialized. (Spark is supported starting from Hive 1.3.0, with HIVE-11180.)

hive.mapjoin.optimized.hashtable.wbsize

- Default Value: `10485760 (10 * 1024 * 1024)`
- Added In: Hive 0.14.0 with HIVE-6430

Optimized hashtable (see **hive.mapjoin.optimized.hashtable**) uses a chain of buffers to store data. This is one buffer size. Hashtable may be slightly faster if this is larger, but for small joins unnecessary memory will be allocated and then trimmed.

hive.mapjoin.lazy.hashtable

- Default Value: `true`
- Added In: Hive 0.13.0 with HIVE-6418 and HIVE-6188
- Removed In: Hive 1.1.0 with HIVE-9331

Whether a MapJoin hashtable should deserialize values on demand. Depending on how many values in the table the join will actually touch, it can save a lot of memory by not creating objects for rows that are not needed. If all rows are needed, obviously there's no gain.

hive.hashtable.initialCapacity

- Default Value: `100000`
- Added In: Hive 0.7.0 with HIVE-1642

Initial capacity of mapjoin hashtable if statistics are absent, or if **hive.hashtable.key.count.adjustment** is set to 0.

hive.hashtable.key.count.adjustment

- Default Value: `1.0`
- Added In: Hive 0.14.0 with HIVE-7616

Adjustment to mapjoin hashtable size derived from table and column statistics; the estimate of the number of keys is divided by this value. If the value is 0, statistics are not used and **hive.hashtable.initialCapacity** is used instead.

hive.hashtable.loadfactor

- Default Value: `0.75`
- Added In: Hive 0.7.0 with HIVE-1642

In the process of Mapjoin, the key/value will be held in the hashtable. This value means the load factor for the in-memory hashtable.

hive.debug.localtask

- Default Value: `false`
- Added In: Hive 0.7.0 with HIVE-1642

hive.outerjoin.supports.filters

- Default Value: `true`
- Added In: Hive 0.7.0 with HIVE-1534
- Removed In: Hive 2.2.0 with HIVE-14522

hive.optimize.skewjoin

- Default Value: `false`
- Added In: Hive 0.6.0

Whether to enable skew join optimization. (Also see **hive.optimize.skewjoin.compiletime**.)

hive.skewjoin.key

- Default Value: `100000`
- Added In: Hive 0.6.0

Determine if we get a skew key in join. If we see more than the specified number of rows with the same key in join operator, we think the key as a skew join key.

hive.skewjoin.mapjoin.map.tasks

- Default Value: `10000`
- Added In: Hive 0.6.0

Determine the number of map task used in the follow up map join job for a skew join. It should be used together with **hive.skewjoin.mapjoin.min.split** to perform a fine grained control.

hive.skewjoin.mapjoin.min.split

- Default Value: `33554432`
- Added In: Hive 0.6.0

Determine the number of map task at most used in the follow up map join job for a skew join by specifying the minimum split size. It should be used together with **hive.skewjoin.mapjoin.map.tasks** to perform a fine grained control.

hive.optimize.skewjoin.compiletime

- Default Value: `false`
- Added In: Hive 0.10.0

Whether to create a separate plan for skewed keys for the tables in the join. This is based on the skewed keys stored in the metadata. At compile time, the plan is broken into different joins: one for the skewed keys, and the other for the remaining keys. And then, a union is performed for the two joins generated above. So unless the same skewed key is present in both the joined tables, the join for the skewed key will be performed as a map-side join.

The main difference between this paramater and **hive.optimize.skewjoin** is that this parameter uses the skew information stored in the metastore to optimize the plan at compile time itself. If there is no skew information in the metadata, this parameter will not have any effect.
Both **hive.optimize.skewjoin.compiletime** and **hive.optimize.skewjoin** should be set to true. (Ideally, **hive.optimize.skewjoin** should be renamed as ***hive.optimize.skewjoin.runtime***, but for backward compatibility that has not been done.)

If the skew information is correctly stored in the metadata, **hive.optimize.skewjoin.compiletime** will change the query plan to take care of it, and **hive.optimize.skewjoin** will be a no-op.

hive.optimize.union.remove

- Default Value: `false`
- Added In: Hive 0.10.0 with HIVE-3276

Whether to remove the union and push the operators between union and the filesink above union. This avoids an extra scan of the output by union. This is independently useful for union queries, and especially useful when **hive.optimize.skewjoin.compiletime** is set to true, since an extra union is inserted.

The merge is triggered if either of **hive.merge.mapfiles** or **hive.merge.mapredfiles** is set to true. If the user has set **hive.merge.mapfiles** to true and **hive.merge.mapredfiles** to false, the idea was that the number of reducers are few, so the number of files anyway is small. However, with this optimization, we are increasing the number of files possibly by a big margin. So, we merge aggresively.

hive.mapred.supports.subdirectories

- Default Value: `false`
- Added In: Hive 0.10.0 with HIVE-3276

Whether the version of Hadoop which is running supports sub-directories for tables/partitions. Many Hive optimizations can be applied if the Hadoop version supports sub-directories for tables/partitions. This support was added by MAPREDUCE-1501.

hive.mapred.mode

- Default Value:
  - Hive 0.x: `nonstrict`
  - Hive 1.x: `nonstrict`
  - Hive 2.x: `strict` (HIVE-12413)
- Added In: Hive 0.3.0

The mode in which the Hive operations are being performed. In `strict` mode, some risky queries are not allowed to run. For example, full table scans are prevented (see HIVE-10454) and ORDER BY requires a LIMIT clause.

hive.exec.script.maxerrsize

- Default Value: `100000`
- Added In: Hive 0.2.0

Maximum number of bytes a script is allowed to emit to standard error (per map-reduce task). This prevents runaway scripts from filling logs partitions to capacity.

hive.script.auto.progress

- Default Value: `false`
- Added In: Hive 0.4.0

Whether Hive Tranform/Map/Reduce Clause should automatically send progress information to TaskTracker to avoid the task getting killed because of inactivity. Hive sends progress information when the script is outputting to stderr. This option removes the need of periodically producing stderr messages, but users should be cautious because this may prevent infinite loops in the scripts to be killed by TaskTracker.

hive.exec.script.allow.partial.consumption

- Default Value: `false`
- Added In: Hive 0.5.0

When enabled, this option allows a user script to exit successfully without consuming all the data from the standard input.

hive.script.operator.id.env.var

- Default Value: `HIVE_SCRIPT_OPERATOR_ID`
- Added In: Hive 0.5.0

Name of the environment variable that holds the unique script operator ID in the user's transform function (the custom mapper/reducer that the user has specified in the query).

hive.script.operator.env.blacklist

- Default Value: `hive.txn.valid.txns,hive.script.operator.env.blacklist`
- Added In: Hive 0.14.0 with HIVE-8341

By default all values in the HiveConf object are converted to environment variables of the same name as the key (with '.' (dot) converted to '_' (underscore)) and set as part of the script operator's environment. However, some values can grow large or are not amenable to translation to environment variables. This value gives a comma separated list of configuration values that will not be set in the environment when calling a script operator. By default the valid transaction list is excluded, as it can grow large and is sometimes compressed, which does not translate well to an environment variable.

Also see:

- **SerDes** for more **hive.script.*** configuration properties

hive.exec.compress.output

- Default Value: `false`
- Added In: Hive 0.2.0

This controls whether the final outputs of a query (to a local/hdfs file or a Hive table) is compressed. The compression codec and other options are determined from Hadoop configuration variables mapred.output.compress* .

hive.exec.compress.intermediate

- Default Value: `false`
- Added In: Hive 0.2.0

This controls whether intermediate files produced by Hive between multiple map-reduce jobs are compressed. The compression codec and other options are determined from Hadoop configuration variables mapred.output.compress*.

hive.exec.parallel

- Default Value: `false`

- Added In: Hive 0.5.0

Whether to execute jobs in parallel. Applies to MapReduce jobs that can run in parallel, for example jobs processing different source tables before a join. As of Hive 0.14, also applies to move tasks that can run in parallel, for example moving files to insert targets during multi-insert.

hive.exec.parallel.thread.number

- Default Value: `8`
- Added In: Hive 0.6.0

How many jobs at most can be executed in parallel.

hive.exec.rowoffset

- Default Value: `false`
- Added In: Hive 0.8.0

Whether to provide the row offset virtual column.

hive.task.progress

- Default Value: `false`
- Added In: Hive 0.5.0
- Removed In: Hive 0.13.0 with HIVE-4518

Whether Hive should periodically update task progress counters during execution. Enabling this allows task progress to be monitored more closely in the job tracker, but may impose a performance penalty. This flag is automatically set to true for jobs with **hive.exec.dynamic.partition** set to true. (This configuration property was removed in release 0.13.0.)

hive.counters.group.name

- Default Value: `HIVE`
- Added In: Hive 0.13.0 with HIVE-4518

Counter group name for counters used during query execution. The counter group is used for internal Hive variables (CREATED_FILE, FATAL_ERROR, and so on).

hive.exec.pre.hooks

- Default Value: (empty)
- Added In: Hive 0.4.0

Comma-separated list of pre-execution hooks to be invoked for each statement. A pre-execution hook is specified as the name of a Java class which implements the org.apache.hadoop.hive.ql.hooks.ExecuteWithHookContext interface.

hive.exec.post.hooks

- Default Value: (empty)
- Added In: Hive 0.5.0

Comma-separated list of post-execution hooks to be invoked for each statement. A post-execution hook is specified as the name of a Java class which implements the org.apache.hadoop.hive.ql.hooks.ExecuteWithHookContext interface.

hive.exec.failure.hooks

- Default Value: (empty)
- Added In: Hive 0.8.0

Comma-separated list of on-failure hooks to be invoked for each statement. An on-failure hook is specified as the name of Java class which implements the org.apache.hadoop.hive.ql.hooks.ExecuteWithHookContext interface.

hive.merge.mapfiles

- Default Value: `true`
- Added In: Hive 0.4.0

Merge small files at the end of a map-only job.

hive.merge.mapredfiles

- Default Value: `false`
- Added In: Hive 0.4.0

Merge small files at the end of a map-reduce job.

hive.mergejob.maponly

- Default Value: `true`
- Added In: Hive 0.6.0
- Removed In: Hive 0.11.0

Try to generate a map-only job for merging files if CombineHiveInputFormat is supported. (This configuration property was removed in release 0.11.0.)

hive.merge.size.per.task

- Default Value: `256000000`
- Added In: Hive 0.4.0

Size of merged files at the end of the job.

hive.merge.smallfiles.avgsize

- Default Value: `16000000`
- Added In: Hive 0.5.0

When the average output file size of a job is less than this number, Hive will start an additional map-reduce job to merge the output files into bigger files. This is only done for map-only jobs if hive.merge.mapfiles is true, and for map-reduce jobs if hive.merge.mapredfiles is true.

hive.heartbeat.interval

- Default Value: `1000`
- Added In: Hive 0.4.0

Send a heartbeat after this interval – used by mapjoin and filter operators.

hive.auto.convert.join

- Default Value: `false` in 0.7.0 to 0.10.0; `true` in 0.11.0 and later (HIVE-3297)
- Added In: 0.7.0 with HIVE-1642

Whether Hive enables the optimization about converting common join into mapjoin based on the input file size. (Note that hive-default.xml.template incorrectly gives the default as false in Hive 0.11.0 through 0.13.1.)

hive.auto.convert.join.noconditionaltask

- Default Value: `true`
- Added In: 0.11.0 with HIVE-3784 (default changed to true with HIVE-4146)

Whether Hive enables the optimization about converting common join into mapjoin based on the input file size. If this parameter is on, and the sum of size for n-1 of the tables/partitions for an n-way join is smaller than the size specified by hive.auto.convert.join.noconditionaltask.size, the join is directly converted to a mapjoin (there is no conditional task).

hive.auto.convert.join.noconditionaltask.size

- Default Value: `10000000`
- Added In: 0.11.0 with HIVE-3784

If hive.auto.convert.join.noconditionaltask is off, this parameter does not take effect. However, if it is on, and the sum of size for n-1 of the tables/partitions for an n-way join is smaller than this size, the join is directly converted to a mapjoin (there is no conditional task). The default is 10MB.

hive.auto.convert.join.use.nonstaged

- Default Value: `false`
- Added In: 0.13.0 with HIVE-6144 (default originally `true`, but changed to `false` with HIVE-6749 also in 0.13.0)

For conditional joins, if input stream from a small alias can be directly applied to the join operator without filtering or projection, the alias need not be pre-staged in the distributed cache via a mapred local task. Currently, this is not working with vectorization or Tez execution engine.

hive.merge.nway.joins

- Default Value: `true`
- Added In: 2.2.0 with HIVE-15655

For multiple joins on the same condition, merge joins together into a single join operator. This is useful in the case of large shuffle joins to avoid a reshuffle phase. Disabling this in Tez will often provide a faster join algorithm in case of left outer joins or a general Snowflake schema.

hive.udtf.auto.progress

- Default Value: `false`
- Added In: Hive 0.5.0

Whether Hive should automatically send progress information to TaskTracker when using UDTF's to prevent the task getting killed because of inactivity. Users should be cautious because this may prevent TaskTracker from killing tasks with infinite loops.

hive.mapred.reduce.tasks.speculative.execution

- Default Value: `true`
- Added In: Hive 0.5.0

Whether speculative execution for reducers should be turned on.

hive.exec.counters.pull.interval

- Default Value: `1000`
- Added In: Hive 0.6.0

The interval with which to poll the JobTracker for the counters the running job. The smaller it is the more load there will be on the jobtracker, the higher it is the less granular the caught will be.

hive.enforce.bucketing

- Default Value:
  - Hive 0.x: `false`
  - Hive 1.x: `false`
  - Hive 2.x: removed, which effectively makes it always true (HIVE-12331)
- Added In: Hive 0.6.0

Whether bucketing is enforced. If `true`, while inserting into the table, bucketing is enforced.

Set to `true` to support INSERT ... VALUES, UPDATE, and DELETE transactions in Hive 0.14.0 and 1.x.x. For a complete list of parameters required for turning on Hive transactions, see **hive.txn.manager**.

hive.enforce.sorting

- Default Value:
  - Hive 0.x: `false`
  - Hive 1.x: `false`
  - Hive 2.x: removed, which effectively makes it always true (HIVE-12331)
- Added In: Hive 0.6.0

Whether sorting is enforced. If true, while inserting into the table, sorting is enforced.

hive.optimize.bucketingsorting

- Default Value: `true`
- Added In: Hive 0.11.0 with HIVE-4240

If **hive.enforce.bucketing** or **hive.enforce.sorting** is true, don't create a reducer for enforcing bucketing/sorting for queries of the form:

```
insert overwrite table T2 select * from T1;
```

where T1 and T2 are bucketed/sorted by the same keys into the same number of buckets. (In Hive 2.0.0 and later, this parameter does not depend on **hive.enforce.bucketing** or **hive.enforce.sorting**.)

hive.optimize.reducededuplication

- Default Value: `true`
- Added In: Hive 0.6.0

Remove extra map-reduce jobs if the data is already clustered by the same key which needs to be used again. This should always be set to true. Since it is a new feature, it has been made configurable.

hive.optimize.reducededuplication.min.reducer

- Default Value: `4`
- Added In: Hive 0.11.0 with HIVE-2340

Reduce deduplication merges two RSs (*reduce sink operators*) by moving key/parts/reducer-num of the child RS to parent RS. That means if reducer-num of the child RS is fixed (order by or forced bucketing) and small, it can make very slow, single MR. The optimization will be disabled if number of reducers is less than specified value.

hive.optimize.correlation

- Default Value: `false`
- Added In: Hive 0.12.0 with HIVE-2206

Exploit intra-query correlations. For details see the Correlation Optimizer design document.

hive.optimize.limittranspose

- Default Value: `false`
- Added In: Hive 2.0.0 with HIVE-11684, modified by HIVE-11775

Whether to push a limit through left/right outer join or union. If the value is true and the size of the outer input is reduced enough (as specified in **hive.optimize.limittranspose.reductionpercentage** and **hive.optimize.limittranspose.reductiontuples**), the limit is pushed to the outer input or union; to remain semantically correct, the limit is kept on top of the join or the union too.

hive.optimize.limittranspose.reductionpercentage

- Default Value: `1.0`
- Added In: Hive 2.0.0 with HIVE-11684, modified by HIVE-11775

When **hive.optimize.limittranspose** is true, this variable specifies the minimal percentage (fractional) reduction of the size of the outer input of the join or input of the union that the optimizer should get in order to apply the rule.

hive.optimize.limittranspose.reductiontuples

- Default Value: `0`
- Added In: Hive 2.0.0 with HIVE-11684, modified by HIVE-11775

When **hive.optimize.limittranspose** is true, this variable specifies the minimal reduction in the number of tuples of the outer input of the join or input of the union that the optimizer should get in order to apply the rule.

hive.optimize.filter.stats.reduction

- Default Value: `false`
- Added In: Hive 2.1.0 with HIVE-13269

Whether to simplify comparison expressions in filter operators using column stats.

hive.optimize.sort.dynamic.partition

- Default Value: `true` in Hive 0.13.0 and 0.13.1; `false` in Hive 0.14.0 and later (HIVE-8151)
- Added In: Hive 0.13.0 with HIVE-6455

When enabled, dynamic partitioning column will be globally sorted. This way we can keep only one record writer open for each partition value in the reducer thereby reducing the memory pressure on reducers.

hive.cbo.enable

- Default Value: `false` in Hive 0.14.*; `true` in Hive 1.1.0 and later (HIVE-8395)
- Added In: Hive 0.14.0 with HIVE-5775 and HIVE-7946

When true, the cost based optimizer, which uses the Calcite framework, will be enabled.

hive.cbo.returnpath.hiveop

- Default Value: `false`
- Added In: Hive 1.2.0 with HIVE-9581 and HIVE-9795

When true, this optimization to CBO Logical plan will add rule to introduce *not null* filtering on join keys. Controls Calcite plan to Hive operator conversion. Overrides hive.optimize.remove.identity.project when set to false.

hive.cbo.cnf.maxnodes

- Default Value: `-1`
- Added In: Hive 2.1.1 with HIVE-14021

When converting to conjunctive normal form (CNF), fail if the expression exceeds the specified threshold; the threshold is expressed in terms of the number of nodes (leaves and interior nodes). The default, -1, does not set up a threshold.

hive.optimize.null.scan

- Default Value: `true`
- Added In: Hive 0.14.0 with HIVE-7385

When true, this optimization will try to not scan any rows from tables which can be determined at query compile time to not generate any rows (e.g., where 1 = 2, where false, limit 0 etc.).

hive.exec.dynamic.partition

- Default Value: `false` prior to Hive 0.9.0; `true` in Hive 0.9.0 and later (HIVE-2835)
- Added In: Hive 0.6.0

Whether or not to allow dynamic partitions in DML/DDL.

hive.exec.dynamic.partition.mode

- Default Value: `strict`
- Added In: Hive 0.6.0

In `strict` mode, the user must specify at least one static partition in case the user accidentally overwrites all partitions. In `nonstrict` mode all partitions are allowed to be dynamic.

Set to `nonstrict` to support INSERT ... VALUES, UPDATE, and DELETE transactions (Hive 0.14.0 and later). For a complete list of parameters required for turning on Hive transactions, see **hive.txn.manager**.

hive.exec.max.dynamic.partitions

- Default Value: `1000`
- Added In: Hive 0.6.0

Maximum number of dynamic partitions allowed to be created in total.

hive.exec.max.dynamic.partitions.pernode

- Default Value: `100`
- Added In: Hive 0.6.0

Maximum number of dynamic partitions allowed to be created in each mapper/reducer node.

hive.exec.max.created.files

- Default Value: `100000`
- Added In: Hive 0.7.0

Maximum number of HDFS files created by all mappers/reducers in a MapReduce job.

hive.exec.default.partition.name

- Default Value: `__HIVE_DEFAULT_PARTITION__`
- Added In: Hive 0.6.0

The default partition name in case the dynamic partition column value is null/empty string or any other values that cannot be escaped. This value must not contain any special character used in HDFS URI (e.g., ':', '%', '/' etc). The user has to be aware that the dynamic partition value should not contain this value to avoid confusions.

hive.fetch.output.serde

- Default Value: `org.apache.hadoop.hive.serde2.DelimitedJSONSerDe`
- Added In: Hive 0.7.0

The SerDe used by FetchTask to serialize the fetch output.

hive.exec.mode.local.auto

- Default Value: `false`
- Added In: Hive 0.7.0 with HIVE-1408

Lets Hive determine whether to run in local mode automatically.

hive.exec.mode.local.auto.inputbytes.max

- Default Value: `134217728`
- Added In: Hive 0.7.0 with HIVE-1408

When **hive.exec.mode.local.auto** is true, input bytes should be less than this for local mode.

hive.exec.mode.local.auto.tasks.max

- Default Value: `4`
- Added In: Hive 0.7.0 with HIVE-1408
- Removed In: Hive 0.9.0 with HIVE-2651

When **hive.exec.mode.local.auto** is true, the number of tasks should be less than this for local mode. Replaced in Hive 0.9.0 by **hive.exec.mode.local.auto.input.files.max**.

hive.exec.mode.local.auto.input.files.max

- Default Value: `4`
- Added In: Hive 0.9.0 with HIVE-2651

When **hive.exec.mode.local.auto** is true, the number of tasks should be less than this for local mode.

hive.exec.drop.ignorenonexistent

- Default Value: `true`
- Added In: Hive 0.7.0 with HIVE-1856 and HIVE-1858

Do not report an error if DROP TABLE/VIEW/PARTITION/INDEX/TEMPORARY FUNCTION specifies a non-existent table/view. Also applies to permanent functions as of Hive 0.13.0.

hive.exec.show.job.failure.debug.info

- Default Value: `true`
- Added In: Hive 0.7.0

If a job fails, whether to provide a link in the CLI to the task with the most failures, along with debugging hints if applicable.

hive.auto.progress.timeout

- Default Value: `0`
- Added In: Hive 0.7.0

How long to run autoprogressor for the script/UDTF operators (in seconds). Set to 0 for forever.

hive.table.parameters.default

- Default Value: (empty)
- Added In: Hive 0.7.0

Default property values for newly created tables.

hive.variable.substitute

- Default Value: `true`
- Added In: Hive 0.7.0

This enables substitution using syntax like `${var}` `${system:var}` and `${env:var}`.

hive.error.on.empty.partition

- Default Value: `false`
- Added In: Hive 0.7.0

Whether to throw an exception if dynamic partition insert generates empty results.

hive.exim.uri.scheme.whitelist

- Default Value: `hdfs,pfile` prior to Hive 2.2.0; `hdfs,pfile,file` in Hive 2.2.0 and later
- Added In: Hive 0.8.0 with HIVE-1918; default changed in Hive 2.2.0 with HIVE-15151

A comma separated list of acceptable URI schemes for import and export.

hive.limit.row.max.size

- Default Value: `100000`
- Added In: Hive 0.8.0

When trying a smaller subset of data for simple LIMIT, how much size we need to guarantee each row to have at least.

hive.limit.optimize.limit.file

- Default Value: `10`
- Added In: Hive 0.8.0

When trying a smaller subset of data for simple LIMIT, maximum number of files we can sample.

hive.limit.optimize.enable

- Default Value: `false`
- Added In: Hive 0.8.0

Whether to enable to optimization to trying a smaller subset of data for simple LIMIT first.

hive.limit.optimize.fetch.max

- Default Value: `50000`
- Added In: Hive 0.8.0

Maximum number of rows allowed for a smaller subset of data for simple LIMIT, if it is a fetch query. Insert queries are not restricted by this limit.

hive.rework.mapredwork

- Default Value: `false`
- Added In: Hive 0.8.0

Should rework the mapred work or not. This is first introduced by SymlinkTextInputFormat to replace symlink files with real paths at compile time.

hive.sample.seednumber

- Default Value: `0`
- Added In: Hive 0.8.0

A number used to percentage sampling. By changing this number, user will change the subsets of data sampled.

hive.autogen.columnalias.prefix.label

- Default Value: `_c`
- Added In: Hive 0.8.0

String used as a prefix when auto generating column alias. By default the prefix label will be appended with a column position number to form the column alias. Auto generation would happen if an aggregate function is used in a select clause without an explicit alias.

hive.autogen.columnalias.prefix.includefuncname

- Default Value: `false`
- Added In: Hive 0.8.0

Whether to include function name in the column alias auto generated by Hive.

hive.exec.perf.logger

- Default Value: `org.apache.hadoop.hive.ql.log.PerfLogger`
- Added In: Hive 0.8.0

The class responsible logging client side performance metrics. Must be a subclass of org.apache.hadoop.hive.ql.log.PerfLogger.

hive.start.cleanup.scratchdir

- Default Value: `false`
- Added In: Hive 0.8.1 with HIVE-2181
- Fixed In: Hive 1.3.0 with HIVE-10415

To clean up the Hive scratch directory while starting the Hive server (or HiveServer2). This is not an option for a multi-user environment since it will accidentally remove the scratch directory in use.

hive.scratchdir.lock

- Default Value: `false`
- Added In: Hive 1.3.0 and 2.1.0 (but not 2.0.x) with HIVE-13429

When true, holds a lock file in the scratch directory. If a Hive process dies and accidentally leaves a dangling scratchdir behind, the cleardanglingscratchdir tool will remove it.

When false, does not create a lock file and therefore the cleardanglingscratchdir tool cannot remove any dangling scratch directories.

hive.output.file.extension

- Default Value: (empty)
- Added In: Hive 0.8.1

String used as a file extension for output files. If not set, defaults to the codec extension for text files (e.g. ".gz"), or no extension otherwise.

hive.insert.into.multilevel.dirs

- Default Value: `false`
- Added In: Hive 0.8.1

Whether to insert into multilevel nested directories like "insert directory '/HIVEFT25686/chinna/' from table".

The following error may be shown when inserting into a nested directory that does not exist:
ERROR org.apache.hadoop.hive.ql.exec.Task: Failed with exception Unable to rename: <xxxx>

To enable automatic subdirectory generation set 'hive.insert.into.multilevel.dirs=true'

hive.conf.validation

- Default Value: `true`
- Added In: Hive 0.10.0 with HIVE-2848

Enables type checking for registered Hive configurations.

As of Hive 0.14.0 (HIVE-7211), a configuration name that starts with "hive." is regarded as a Hive system property. With **hive.conf.validation** true (default), any attempts to set a configuration property that starts with "hive." which is not registered to the Hive system will throw an exception.

hive.fetch.task.conversion

- Default Value: `minimal` in Hive 0.10.0 through 0.13.1, `more` in Hive 0.14.0 and later
- Added In: Hive 0.10.0 with HIVE-2925; default changed in Hive 0.14.0 with HIVE-7397

Some select queries can be converted to a single FETCH task, minimizing latency. Currently the query should be single sourced not having any subquery and should not have any aggregations or distincts (which incur RS – ReduceSinkOperator, requiring a MapReduce task), lateral views and joins.

Supported values are none, `minimal` and `more`.

0. `none`: Disable hive.fetch.task.conversion (value added in Hive 0.14.0 with HIVE-8389)
1. `minimal`: SELECT *, FILTER on partition columns (WHERE and HAVING clauses), LIMIT only
2. `more`: SELECT, FILTER, LIMIT only (including TABLESAMPLE, virtual columns)

   "`more`" can take any kind of expressions in the SELECT clause, including UDFs.
   (UDTFs and lateral views are not yet supported – see HIVE-5718.)

hive.map.groupby.sorted

- Default Value:
  - Hive 0.x and 1.x: `false`
  - Hive 2.0 and later: `true` (HIVE-12325)
- Added In: Hive 0.10.0 with HIVE-3432

If the bucketing/sorting properties of the table exactly match the grouping key, whether to perform the group by in the mapper by using BucketizedHiveInputFormat. The only downside to this is that it limits the number of mappers to the number of files.

hive.map.groupby.sorted.testmode

- Default Value: `false`
- Added In: Hive 0.11.0 with HIVE-4281
- Removed In: Hive 2.0.0 with HIVE-12325

If the bucketing/sorting properties of the table exactly match the grouping key, whether to perform the group by in the mapper by using BucketizedHiveInputFormat. If the test mode is set, the plan is not converted, but a query property is set to denote the same. (This configuration property was removed in release 2.0.0.)

hive.groupby.orderby.position.alias

- Default Value: `false`
- Added In: Hive 0.11.0 with HIVE-581
- Deprecated In: Hive 2.2.0 with HIVE-15797

Whether to enable using Column Position Alias in GROUP BY and ORDER BY clauses of queries (deprecated as of Hive 2.2.0; use hive.groupby.position.alias and hive.orderby.position.alias instead).

hive.groupby.position.alias

- Default Value: `false`
- Added In: Hive 2.2.0 with HIVE-15797

Whether to enable using Column Position Alias in GROUP BY.

hive.orderby.position.alias

- Default Value: `true`
- Added In: Hive 2.2.0 with HIVE-15797

Whether to enable using Column Position Alias in ORDER BY.

hive.fetch.task.aggr

- Default Value: `false`
- Added In: Hive 0.12.0 with HIVE-4002 (description added in Hive 0.13.0 with HIVE-5793)

Aggregation queries with no group-by clause (for example, `select count(*) from src`) execute final aggregations in a single reduce task. If this parameter is set to `true`, Hive delegates the final aggregation stage to a fetch task, possibly decreasing the query time.

hive.fetch.task.conversion.threshold

- Default Value: `-1` in Hive 0.13.0 and 0.13.1, `1073741824` (1 GB) in Hive 0.14.0 and later
- Added In: Hive 0.13.0 with HIVE-3990; default changed in Hive 0.14.0 with HIVE-7397

Input threshold (in bytes) for applying **hive.fetch.task.conversion**. If target table is native, input length is calculated by summation of file lengths. If it's not native, the storage handler for the table can optionally implement the org.apache.hadoop.hive.ql.metadata.InputEstimator interface. A negative threshold means **hive.fetch.task.conversion** is applied without

any input length threshold.

hive.limit.pushdown.memory.usage

- Default Value: `-1`
- Added In: Hive 0.12.0 with HIVE-3562

The maximum memory to be used for hash in RS operator for top K selection. The default value "-1" means no limit.

hive.cache.expr.evaluation

- Default Value: `true`
- Added In: Hive 0.12.0 with HIVE-4209
- Bug Fix: Hive 0.14.0 with HIVE-7314 (expression caching doesn't work when using UDF inside another UDF or a Hive function)

If true, the evaluation result of a deterministic expression referenced twice or more will be cached. For example, in a filter condition like "... where key + 10 > 10 or key + 10 = 0" the expression "key + 10" will be evaluated/cached once and reused for the following expression ("key + 10 = 0"). Currently, this is applied only to expressions in select or filter operators.

hive.resultset.use.unique.column.names

- Default Value: `true`
- Added In: Hive 0.13.0 with HIVE-6687

Make column names unique in the result set by qualifying column names with table alias if needed. Table alias will be added to column names for queries of type "select *" or if query explicitly uses table alias "select r1.x..".

hive.support.quoted.identifiers

- Default Value: `column`
- Added In: Hive 0.13.0 with HIVE-6013

Whether to use quoted identifiers. Value can be "`none`" or "`column`".

> `column`: Column names can contain any Unicode character. Any column name that is specified within backticks (`) is treated literally. Within a backtick string, use double backticks (``) to represent a backtick character.
> `none`: Only alphanumeric and underscore characters are valid in identifiers. Backticked names are interpreted as regular expressions. This is also the behavior in releases prior to 0.13.0.

hive.plan.serialization.format

- Default Value: `kryo`
- Added In: Hive 0.13.0 with HIVE-1511
- Removed a Value In: Hive 2.0.0 with HIVE-12609 javaXML is no longer supported

Query plan format serialization between client and task nodes. Two supported values are `kryo` and `javaXML` (prior to Hive 2.0.0). Kryo is the default (and starting from Hive 2.0.0 Kryo is the only supported value).

hive.exec.check.crossproducts

- Default Value: `true`
- Added In: Hive 0.13.0 with HIVE-6643

Check if a query plan contains a cross product. If there is one, output a warning to the session's console.

hive.display.partition.cols.separately

- Default Value: `true`
- Added In: Hive 0.13.0 with HIVE-6689

In older Hive versions (0.10 and earlier) no distinction was made between partition columns or non-partition columns while displaying columns in DESCRIBE TABLE. From version 0.12 onwards, they are displayed separately. This flag will let you get the old behavior, if desired. See test-case in patch for HIVE-6689.

hive.limit.query.max.table.partition

- Default Value: `-1`
- Added In: Hive 0.13.0 with HIVE-6492
- Deprecated In: Hive 2.2.0 with HIVE-13884 (See **hive.metastore.limit.partition.request**.)
- Removed In: Hive 3.0.0 with HIVE-17965

To protect the cluster, this controls how many partitions can be scanned for each partitioned table. The default value "-1" means no limit. The limit on partitions does not affect metadata-only queries.

hive.files.umask.value

- Default Value: `0002`
- Added In: (none, but temporarily in patches for HIVE-2504 before release 0.9.0)
- Removed In: Hive 0.9.0 (HIVE-2504-1.patch), replaced by **hive.warehouse.subdir.inherit.perms**

Obsolete: The `dfs.umask` value for the Hive-created folders.

hive.optimize.sampling.orderby

- Default Value: `false`
- Added In: Hive 0.12.0 with HIVE-1402

Uses sampling on order-by clause for parallel execution.

hive.optimize.sampling.orderby.number

- Default Value: `1000`
- Added In: Hive 0.12.0 with HIVE-1402

With hive.optimize.sampling.orderby=true, total number of samples to be obtained to calculate partition keys.

hive.optimize.sampling.orderby.percent

- Default Value: 0.1
- Added In: Hive 0.12.0 with HIVE-1402

With hive.optimize.sampling.orderby=true, probability with which a row will be chosen.

hive.compat

- Default Value: 0.12
- Added In: Hive 0.13.0 with HIVE-6012

Enable (configurable) deprecated behaviors of arithmetic operations by setting the desired level of backward compatibility. The default value gives backward-compatible return types for numeric operations. Other supported release numbers give newer behavior for numeric operations, for example 0.13 gives the more SQL compliant return types introduced in HIVE-5356.

The value "latest" specifies the latest supported level. Currently, this only affects division of integers.

> Setting to 0.12 (default) maintains division behavior in Hive 0.12 and earlier releases: int / int = double.
> Setting to 0.13 gives division behavior in Hive 0.13 and later releases: int / int = decimal.

An invalid setting will cause an error message, and the default support level will be used.

hive.optimize.constant.propagation

- Default Value: `true`
- Added In: Hive 0.14.0 with HIVE-5771

Whether to enable the constant propagation optimizer.

hive.entity.capture.transform

- Default Value: `false`
- Added In: Hive 1.1.0 with HIVE-8938

Enable capturing compiler read entity of transform URI which can be introspected in the semantic and exec hooks.

hive.support.sql11.reserved.keywords

- Default Value: `true`
- Added In: Hive 1.2.0 with HIVE-6617

Whether to enable support for SQL2011 reserved keywords. When enabled, will support (part of) SQL2011 reserved keywords.

hive.log.explain.output

- Default Value: `false`
- Added In: 1.1.0 with HIVE-8600

When enabled, will log EXPLAIN EXTENDED output for the query at log4j INFO level and in HiveServer2 WebUI / Drilldown / Query Plan.

From Hive 3.1.0 onwards, this configuration property only logs to the log4j INFO. To log the EXPLAIN EXTENDED output in WebUI / Drilldown / Query Plan from Hive 3.1.0 onwards, use **hive.server2.webui.explain.output**.

hive.explain.user

- Default Value: `false`
- Added In: Hive 1.2.0 with HIVE-9780

Whether to show explain result at user level. When enabled, will log EXPLAIN output for the query at user level. (Tez only. For Spark, see **hive.spark.explain.user**.)

hive.typecheck.on.insert

- Default Value: `true`
- Added In: Hive 0.12.0 with HIVE-5297 for insert partition
- Extended In: Hive 1.2 with HIVE-10307 for alter, describe partition, etc.

Whether to check, convert, and normalize partition value specified in partition specification to conform to the partition column type.

hive.exec.temporary.table.storage

- Default Value: `default`
- Added In: Hive 1.1.0 with HIVE-7313

Expects one of [`memory`, `ssd`, `default`].

Define the storage policy for temporary tables. Choices between memory, ssd and default. See HDFS Storage Types and Storage Policies.

hive.optimize.distinct.rewrite

- Default Value: `true`
- Added In: Hive 1.2.0 with HIVE-10568

When applicable, this optimization rewrites distinct aggregates from a single-stage to multi-stage aggregation. This may not be optimal in all cases. Ideally, whether to trigger it or not should be a cost-based decision. Until Hive formalizes the cost model for this, this is config driven.

hive.optimize.point.lookup

- Default Value: `true`
- Added In: Hive 2.0.0 with HIVE-11461

Whether to transform OR clauses in Filter operators into IN clauses.

hive.optimize.point.lookup.min

- Default Value: `31`
- Added In: Hive 2.0.0 with HIVE-11573

Minimum number of OR clauses needed to transform into IN clauses.

hive.allow.udf.load.on.demand

- Default Value: `false`
- Added In: Hive 2.1.0 with HIVE-13596

Whether enable loading UDFs from metastore on demand; this is mostly relevant for HS2 and was the default behavior before Hive 1.2.

hive.async.log.enabled

- Default Value: `true`
- Added In: Hive 2.1.0 with HIVE-13027

Whether to enable Log4j2's asynchronous logging. Asynchronous logging can give significant performance improvement as logging will be handled in a separate thread that uses the LMAX disruptor queue for buffering log messages.

Refer to https://logging.apache.org/log4j/2.x/manual/async.html for benefits and drawbacks.

hive.msck.repair.batch.size

- Default Value: 0
- Added In: Hive 2.2.0 with HIVE-12077

To run the MSCK REPAIR TABLE command batch-wise. If there is a large number of untracked partitions, by configuring a value to the property it will execute in batches internally. The default value of the property is zero, which means it will execute all the partitions at once.

hive.exec.copyfile.maxnumfiles

- Default Value: 1
- Added In: Hive 2.3.0 with HIVE-14864

Maximum number of files Hive uses to do sequential HDFS copies between directories. Distributed copies (distcp) will be used instead for larger numbers of files so that copies can be done faster.

hive.exec.copyfile.maxsize

- Default Value: 32 megabytes
- Added In: Hive 1.1.0 with HIVE-8750

Maximum file size (in bytes) that Hive uses to do single HDFS copies between directories. Distributed copies (distcp) will be used instead for bigger files so that copies can be done faster.

hive.exec.stagingdir

- Default Value: `.hive-staging`
- Added in: Hive 1.1.0 with HIVE-8750

Directory name that will be created inside table locations in order to support HDFS encryption. This is replaces `hive.exec.scratchdir` for query results with the exception of read-only tables. In all cases `hive.exec.scratchdir` is still used for other temporary files, such as job plans.

hive.query.lifetime.hooks

- Default Value: (empty)
- Added In: Hive 2.3.0 with HIVE-14340

A comma separated list of hooks which implement QueryLifeTimeHook. These will be triggered before/after query compilation and before/after query execution, in the order specified. As of Hive 3.0.0 (HIVE-16363), this config can be used to specify implementations of QueryLifeTimeHookWithParseHooks. If they are specified then they will be invoked in the same places as QueryLifeTimeHooks and will be invoked during pre and post query parsing.

hive.remove.orderby.in.subquery

- Default Value: `true`
- Added In: Hive 3.0.0 with HIVE-6348

If set to true, order/sort by without limit in subqueries and views will be removed.

SerDes and I/O

SerDes

hive.script.serde

- Default Value: `org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe`
- Added In: Hive 0.4.0

The default SerDe for transmitting input data to and reading output data from the user scripts.

hive.script.recordreader

- Default Value: `org.apache.hadoop.hive.ql.exec.TextRecordReader`
- Added In: Hive 0.4.0

The default record reader for reading data from the user scripts.

hive.script.recordwriter

- Default Value: `org.apache.hadoop.hive.ql.exec.TextRecordWriter`
- Added In: Hive 0.5.0

The default record writer for writing data to the user scripts.

hive.default.serde

- Default Value: `org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe`
- Added in: Hive 0.14 with HIVE-5976

The default SerDe Hive will use for storage formats that do not specify a SerDe. Storage formats that currently do not specify a SerDe include 'TextFile, RcFile'.

See Registration of Native SerDes for more information for storage formats and SerDes.

hive.lazysimple.extended_boolean_literal

- Default Value: `false`
- Added in: Hive 0.14 with HIVE-3635

LazySimpleSerDe uses this property to determine if it treats 'T', 't', 'F', 'f', '1', and '0' as extended, legal boolean literals, in addition to 'TRUE' and 'FALSE'. The default is `false`, which means only 'TRUE' and 'FALSE' are treated as legal boolean literals.

I/O

hive.io.exception.handlers

- Default Value: (empty)
- Added In: Hive 0.8.1

A list of I/O exception handler class names. This is used to construct a list of exception handlers to handle exceptions thrown by record readers.

hive.input.format

- Default Value: `org.apache.hadoop.hive.ql.io.CombineHiveInputFormat`
- Added In: Hive 0.5.0

The default input format. Set this to HiveInputFormat if you encounter problems with CombineHiveInputFormat.

Also see:

- **hive.tez.input.format**

File Formats

hive.default.fileformat

- Default Value: `TextFile`
- Added In: Hive 0.2.0

Default file format for CREATE TABLE statement. Options are TextFile, SequenceFile, RCfile, ORC, and Parquet.

Users can explicitly say CREATE TABLE ... STORED AS TEXTFILE|SEQUENCEFILE|RCFILE|ORC|AVRO|INPUTFORMAT...OUTPUTFORMAT... to override. (RCFILE was added in Hive 0.6.0, ORC in 0.11.0, AVRO in 0.14.0, and Parquet in 2.3.0) See Row Format, Storage Format, and SerDe for details.

hive.default.fileformat.managed

- Default Value: none
- Added In: Hive 1.2.0 with HIVE-9915

Default file format for CREATE TABLE statement applied to managed tables only. External tables will be created with format specified by hive.default.fileformat. Options are none, TextFile, SequenceFile, RCfile, ORC, and Parquet (as of Hive 2.3.0). Leaving this null will result in using hive.default.fileformat for all native tables. For non-native tables the file format is determined by the storage handler, as shown below (see the StorageHandlers section for more information on managed/external and native/non-native terminology).

	Native	Non-Native
Managed	hive.default.fileformat.managed (or fall back to hive.default.fileformat)	Not covered by default file-formats
External	hive.default.fileformat	Not covered by default file-formats

hive.fileformat.check

- Default Value: `true`
- Added In: Hive 0.5.0

Whether to check file format or not when loading data files.

hive.query.result.fileformat

- Default Value:
  - Hive 0.x, 1.x, and 2.0: `TextFile`
  - Hive 2.1 onward: `SequenceFile`
- Added In: Hive 0.7.0 with HIVE-1598

File format to use for a query's intermediate results. Options are TextFile, SequenceFile, and RCfile. Default value is changed to SequenceFile since Hive 2.1.0 (HIVE-1608).

RCFile Format

hive.io.rcfile.record.interval

- Default Value: 2147483647
- Added In: Hive 0.4.0 with HIVE-352; added to HiveConf.java in Hive 0.14.0 with HIVE-7211

hive.io.rcfile.column.number.conf

- Default Value: 0
- Added In: Hive 0.4.0 with HIVE-352; added to HiveConf.java in Hive 0.14.0 with HIVE-7211

hive.io.rcfile.tolerate.corruptions

- Default Value: false
- Added In: Hive 0.4.0 with HIVE-352; added to HiveConf.java in Hive 0.14.0 with HIVE-7211

hive.io.rcfile.record.buffer.size

- Default Value: 4194304
- Added In: Hive 0.4.0 with HIVE-352; added to HiveConf.java in Hive 0.14.0 with HIVE-7211

ORC File Format

The ORC file format was introduced in Hive 0.11.0. See ORC Files for details.

Besides the configuration properties listed in this section, some properties in other sections are also related to ORC:

- hive.default.fileformat
- hive.stats.gather.num.threads

hive.exec.orc.memory.pool

- Default Value: `0.5`
- Added In: Hive 0.11.0 with HIVE-4248

Maximum fraction of heap that can be used by ORC file writers.

hive.exec.orc.write.format

- Default Value: (empty)
- Added In: Hive 0.12.0 with HIVE-4123; default changed from 0.11 to null with HIVE-5091 (also in Hive 0.12.0)

Define the version of the file to write. Possible values are 0.11 and 0.12. If this parameter is not defined, ORC will use the run length encoding (RLE) introduced in Hive 0.12. Any value other than 0.11 results in the 0.12 encoding.

Additional values may be introduced in the future (see HIVE-6002).

hive.exec.orc.base.delta.ratio

- Default Value: `8`
- Added In: Hive 1.3.0 and 2.1.0 with HIVE-13563

Define the ratio of base writer and delta writer in terms of STRIPE_SIZE and BUFFER_SIZE.

hive.exec.orc.default.stripe.size

- Default Value: `256*1024*1024` (268,435,456) in 0.13.0;
  `64*1024*1024` (67,108,864) in 0.14.0
- Added In: Hive 0.13.0 with HIVE-5425; default changed in 0.14.0 with HIVE-7231 and HIVE-7490

Define the default ORC stripe size, in bytes.

hive.exec.orc.default.block.size

- Default Value: `256*1024*1024` (268,435,456)
- Added In: Hive 0.14.0 with HIVE-7231

Define the default file system block size for ORC files.

hive.exec.orc.dictionary.key.size.threshold

- Default Value: `0.8`
- Added In: Hive 0.12.0 with HIVE-4324

If the number of keys in a dictionary is greater than this fraction of the total number of non-null rows, turn off dictionary encoding. Use 1 to always use dictionary encoding.

hive.exec.orc.default.row.index.stride

- Default Value: `10000`
- Added In: Hive 0.13.0 with HIVE-5728

Define the default ORC index stride in number of rows. (*Stride* is the number of rows an index entry represents.)

hive.exec.orc.default.buffer.size

- Default Value: `256*1024` (262,144)
- Added In: Hive 0.13.0 with HIVE-5728

Define the default ORC buffer size, in bytes.

hive.exec.orc.default.block.padding

- Default Value: `true`
- Added In: Hive 0.13.0 with HIVE-5728

Define the default block padding. Block padding was added in Hive 0.12.0 (HIVE-5091, "ORC files should have an option to pad stripes to the HDFS block boundaries").

hive.exec.orc.block.padding.tolerance

- Default Value: `0.05`
- Added In: Hive 0.14.0 with HIVE-7231

Define the tolerance for block padding as a decimal fraction of stripe size (for example, the default value 0.05 is 5% of the stripe size). For the defaults of 64Mb ORC stripe and 256Mb HDFS blocks, a maximum of 3.2Mb will be reserved for padding within the 256Mb block with the default **hive.exec.orc.block.padding.tolerance**. In that case, if the available size within the block is more than 3.2Mb, a new smaller stripe will be inserted to fit within that space. This will make sure that no stripe written will cross block boundaries and cause remote reads within a node local task.

hive.exec.orc.default.compress

- Default Value: `ZLIB`
- Added In: Hive 0.13.0 with HIVE-5728

Define the default compression codec for ORC file.

hive.exec.orc.encoding.strategy

- Default Value: `SPEED`
- Added In: Hive 0.14.0 with HIVE-7219

Define the encoding strategy to use while writing data. Changing this will only affect the light weight encoding for integers. This flag will not change the compression level of higher level compression codec (like ZLIB). Possible options are SPEED and COMPRESSION.

hive.orc.splits.include.file.footer

- Default Value: `false`
- Added In: Hive 0.13.0 with HIVE-6125 and HIVE-6128

If turned on, splits generated by ORC will include metadata about the stripes in the file. This data is read remotely (from the client or HiveServer2 machine) and sent to all the tasks.

hive.orc.cache.stripe.details.size

- Default Value: `10000`
- Added In: Hive 0.13.0 with HIVE-6125 and HIVE-6128

Cache size for keeping meta information about ORC splits cached in the client.

hive.orc.cache.use.soft.references

- Default Value: `false`

- Added In: Hive 1.3.0, Hive 2.1.1, Hive 2.2.0 with HIVE-13985

By default, the cache that ORC input format uses to store the ORC file footer uses hard references for the cached object. Setting this to true can help avoid out-of-memory issues under memory pressure (in some cases) at the cost of slight unpredictability in overall query performance.

hive.io.sarg.cache.max.weight.mb

- Default Value: `10`
- Added In: Hive 2.2.1, Hive 2.3.1, Hive 2.4.0, Hive 3.0.0 with HIVE-17669

The maximum weight allowed for the SearchArgument Cache, in megabytes. By default, the cache allows a max-weight of 10MB, after which entries will be evicted. Set to 0, to disable SearchArgument caching entirely.

hive.orc.compute.splits.num.threads

- Default Value: `10`
- Added In: Hive 0.13.0 with HIVE-6125 and HIVE-6128

How many threads ORC should use to create splits in parallel.

hive.exec.orc.split.strategy

- Default Value: HYBRID
- Added In: Hive 1.2.0 with HIVE-10114

What strategy ORC should use to create splits for execution. The available options are "BI", "ETL" and "HYBRID".

The HYBRID mode reads the footers for all files if there are fewer files than expected mapper count, switching over to generating 1 split per file if the average file sizes are smaller than the default HDFS blocksize. ETL strategy always reads the ORC footers before generating splits, while the BI strategy generates per-file splits fast without reading any data from HDFS.

hive.exec.orc.skip.corrupt.data

- Default Value: `false`
- Added In: Hive 0.13.0 with HIVE-6382

If ORC reader encounters corrupt data, this value will be used to determine whether to skip the corrupt data or throw an exception. The default behavior is to throw an exception.

hive.exec.orc.zerocopy

- Default Value: `false`
- Added In: Hive 0.13.0 with HIVE-6347 and HIVE-6360

Use zerocopy reads with ORC. (This requires Hadoop 2.3 or later.)

hive.merge.orcfile.stripe.level

- Default Value: `true`
- Added In: Hive 0.14.0 with HIVE-7509

When **hive.merge.mapfiles**, **hive.merge.mapredfiles** or **hive.merge.tezfiles** is enabled while writing a table with ORC file format, enabling this configuration property will do stripe-level fast merge for small ORC files. Note that enabling this configuration property will not honor the padding tolerance configuration (**hive.exec.orc.block.padding.tolerance**).

hive.orc.row.index.stride.dictionary.check

- Default Value: `true`
- Added In: Hive 0.14.0 with HIVE-7832

If enabled dictionary check will happen after first row index stride (default 10000 rows) else dictionary check will happen before writing first stripe. In both cases, the decision to use dictionary or not will be retained thereafter.

hive.exec.orc.compression.strategy

- Default Value: `SPEED`
- Added In: Hive 0.14.0 with HIVE-7859

Define the compression strategy to use while writing data. This changes the compression level of higher level compression codec (like ZLIB).

Value can be `SPEED` or `COMPRESSION`.

Parquet

Parquet is supported by a plugin in Hive 0.10, 0.11, and 0.12 and natively in Hive 0.13 and later. See Parquet for details.

hive.parquet.timestamp.skip.conversion

- Default Value: `true`
- Added In: Hive 1.2.0 with HIVE-9482

Current Hive implementation of Parquet stores timestamps in UTC on-file, this flag allows skipping of the conversion on reading Parquet files created from other tools that may not have done so.

Vectorization

Hive added vectorized query execution in release 0.13.0 (HIVE-4160, HIVE-5283). For more information see the design document Vectorized Query Execution.

hive.vectorized.execution.enabled

- Default Value: `false`
- Added In: Hive 0.13.0 with HIVE-5283

This flag should be set to true to enable vectorized mode of query execution. The default value is false.

hive.vectorized.execution.reduce.enabled

- Default Value: `true`
- Added In: Hive 0.14.0 with HIVE-7405

This flag should be set to true to enable vectorized mode of the reduce-side of query execution. The default value is true.

hive.vectorized.execution.reduce.groupby.enabled

- Default Value: `true`
- Added In: Hive 0.14.0 with HIVE-8052

This flag should be set to true to enable vectorized mode of the reduce-side GROUP BY query execution. The default value is true.

hive.vectorized.execution.reducesink.new.enabled

- Default Value: `true`
- Added In: Hive 2.0.0 with HIVE-12290

This flag should be set to true to enable the new vectorization of queries using ReduceSink.

hive.vectorized.execution.mapjoin.native.enabled

- Default Value: `true`
- Added In: Hive 1.2.0 with HIVE-9824

This flag should be set to true to enable native (i.e. non-pass through) vectorization of queries using MapJoin.

hive.vectorized.execution.mapjoin.native.multikey.only.enabled

- Default Value: `false`
- Added In: Hive 1.2.0 with HIVE-9824

This flag should be set to true to restrict use of native vector map join hash tables to the MultiKey in queries using MapJoin.

hive.vectorized.execution.mapjoin.minmax.enabled

- Default Value: `false`
- Added In: Hive 1.2.0 with HIVE-9824

This flag should be set to true to enable vector map join hash tables to use max / max filtering for integer join queries using MapJoin.

hive.vectorized.execution.mapjoin.overflow.repeated.threshold

- Default Value: `-1`
- Added In: Hive 1.2.0 with HIVE-9824

The number of small table rows for a match in vector map join hash tables where we use the repeated field optimization in overflow vectorized row batch for join queries using MapJoin. A value of `-1` means do use the join result optimization. Otherwise, threshold value can be 0 to maximum integer.

hive.vectorized.execution.mapjoin.native.fast.hashtable.enabled

- Default Value: `false`
- Added In: Hive 1.2.0 with HIVE-9824

This flag should be set to true to enable use of native fast vector map join hash tables in queries using MapJoin.

hive.vectorized.groupby.checkinterval

- Default Value: `100000`
- Added In: Hive 0.13.0 with HIVE-5692

Number of entries added to the GROUP BY aggregation hash before a recomputation of average entry size is performed.

hive.vectorized.groupby.maxentries

- Default Value: `1000000`
- Added In: Hive 0.13.0 with HIVE-5692

Maximum number of entries in the vector GROUP BY aggregation hashtables. Exceeding this will trigger a flush regardless of memory pressure condition.

hive.vectorized.use.vectorized.input.format

- Default Value: `true`
- Added In: Hive 2.1.0 with HIVE-12878

This flag should be set to true to allow Hive to take advantage of input formats that support vectorization. The default value is true.

hive.vectorized.use.vector.serde.deserialize

- Default Value: `false`
- Added In: Hive 2.1.0 with HIVE-12878

This flag should be set to true to enable vectorizing rows using vector deserialize. The default value is false.

hive.vectorized.use.row.serde.deserialize

- Default Value: `false`
- Added In: Hive 2.1.0 with HIVE-12878

This flag should be set to true to enable vectorizing using row deserialize. The default value is false.

hive.vectorized.input.format.excludes

- Default Value: (empty)
- Added in: Hive 2.4.0 with HIVE-17534

This flag should be used to provide a comma separated list of fully qualified classnames to exclude certain FileInputFormats from vectorized execution using the vectorized file inputformat. Note that vectorized execution could still occur for that input format based on whether **hive.vectorized.use.vector.serde.deserialize** or **hive.vectorized.use.row.serde.deserialize** is enabled or not.

MetaStore

In addition to the Hive metastore properties listed in this section, some properties are listed in other sections:

- Hive Metastore Security
  - **hive.metastore.pre.event.listeners**

    - **hive.security.metastore.authorization.manager**
    - **hive.security.metastore.authenticator.manager**
    - **hive.security.metastore.authorization.auth.reads**
- Metrics
    - **hive.metastore.metrics.enabled**

hive.metastore.local

- Default Value: `true`
- Added In: Hive 0.8.1
- Removed In: Hive 0.10 with HIVE-2585

Controls whether to connect to remote metastore server or open a new metastore server in Hive Client JVM. As of Hive 0.10 this is no longer used. Instead if **`hive.metastore.uris`** is set then `remote` mode is assumed otherwise `local`.

javax.jdo.option.ConnectionURL

- Default Value: `jdbc:derby:;databaseName=metastore_db;create=true`
- Added In: Hive 0.6.0

JDBC connect string for a JDBC metastore.

javax.jdo.option.ConnectionDriverName

- Default Value: `org.apache.derby.jdbc.EmbeddedDriver`
- Added In: Hive 0.8.1

Driver class name for a JDBC metastore.

javax.jdo.PersistenceManagerFactoryClass

- Default Value: `org.datanucleus.jdo.JDOPersistenceManagerFactory`
- Added In: Hive 0.8.1

Class implementing the JDO PersistenceManagerFactory.

javax.jdo.option.DetachAllOnCommit

- Default Value: `true`
- Added In: Hive 0.8.1

Detaches all objects from session so that they can be used after transaction is committed.

javax.jdo.option.NonTransactionalRead

- Default Value: `true`
- Added In: Hive 0.8.1

Reads outside of transactions.

javax.jdo.option.ConnectionUserName

- Default Value: `APP`
- Added In: Hive 0.8.1

Username to use against metastore database.

javax.jdo.option.ConnectionPassword

- Default Value: `mine`
- Added In: Hive 0.3.0

Password to use against metastore database.

For an alternative configuration, see Removing Hive Metastore Password from Hive Configuration.

javax.jdo.option.Multithreaded

- Default Value: `true`
- Added In: Hive 0.8.0

Set this to true if multiple threads access metastore through JDO concurrently.

datanucleus.connectionPoolingType

- Default Value: DBCP in Hive 0.7 to 0.11; BoneCP in 0.12 to 2.3; HikariCP in 3.0 and later
- Added In: Hive 0.7.0

Uses a HikariCP connection pool for JDBC metastore from 3.0 release onwards (HIVE-16383).

Uses a BoneCP connection pool for JDBC metastore in release 0.12 to 2.3 (HIVE-4807), or a DBCP connection pool in releases 0.7 to 0.11.

As of Hive 2.2.0 (HIVE-13159), this parameter can also be set to `none`.

datanucleus.connectionPool.maxPoolSize

- Default Value: 10
- Added In: Hive 3.0.0

Specify the maximum number of connections in the connection pool.

Note: The configured size will be used by 2 connection pools (TxnHandler and ObjectStore).

When configuring the max connection pool size, it is recommended to take into account the number of metastore instances and the number of HiveServer2 instances

configured with embedded metastore. To get optimal performance, set config to meet the following condition

(2 * pool_size * metastore_instances + 2 * pool_size * HS2_instances_with_embedded_metastore) = (2 * physical_core_count + hard_disk_count).

datanucleus.validateTables

- Default Value: `false`
- Added In: Hive 0.7.0
- Removed In: Hive 2.0.0 with HIVE-6113, replaced by **datanucleus.schema.validateTables**

Validates existing schema against code. Turn this on if you want to verify existing schema.

datanucleus.schema.validateTables

- Default Value: `false`
- Added In: Hive 2.0.0 with HIVE-6113, replaces **datanucleus.validateTables**

Validates existing schema against code. Turn this on if you want to verify existing schema.

datanucleus.validateColumns

- Default Value: `false`
- Added In: Hive 0.7.0
- Removed In: Hive 2.0.0 with HIVE-6113, replaced by **datanucleus.schema.validateColumns**

Validates existing schema against code. Turn this on if you want to verify existing schema.

datanucleus.schema.validateColumns

- Default Value: `false`
- Added In: Hive 2.0.0 with HIVE-6113, replaces **datanucleus.validateColumns**

Validates existing schema against code. Turn this on if you want to verify existing schema.

datanucleus.validateConstraints

- Default Value: `false`
- Added In: Hive 0.7.0
- Removed In: Hive 2.0.0 with HIVE-6113, replaced by **datanucleus.schema.validateConstraints**

Validates existing schema against code. Turn this on if you want to verify existing schema.

datanucleus.schema.validateConstraints

- Default Value: `false`
- Added In: Hive 2.0.0 with HIVE-6113, replaces **datanucleus.validateConstraints**

Validates existing schema against code. Turn this on if you want to verify existing schema.

datanucleus.storeManagerType

- Default Value: `rdbms`
- Added In: Hive 0.7.0

Metadata store type.

datanucleus.fixedDatastore

- Default Value:
  - Hive 0.x: `false`
  - Hive 1.x: `false`
- Added In: Hive 0.12.0 with HIVE-3764
- Removed In: Hive 2.0.0 with HIVE-6113

Dictates whether to allow updates to schema or not.

datanucleus.autoCreateSchema

- Default Value: `true`
- Added In: Hive 0.7.0
- Removed In: Hive 2.0.0 with HIVE-6113, replaced by **datanucleus.schema.autoCreateAll**

Creates necessary schema on a startup if one does not exist. Set this to false, after creating it once.

In Hive 0.12.0 and later releases, **datanucleus.autoCreateSchema** is disabled if **hive.metastore.schema.verification** is `true`.

datanucleus.schema.autoCreateAll

- Default Value: `false`
- Added In: Hive 2.0.0 with HIVE-6113, replaces **datanucleus.autoCreateSchema** (with different default value)

Creates necessary schema on a startup if one does not exist. Reset this to false, after creating it once.

datanucleus.schema.autoCreateAll is disabled if **hive.metastore.schema.verification** is `true`.

datanucleus.autoStartMechanismMode

- Default Value: `checked`
- Added In: Hive 0.7.0

Throw exception if metadata tables are incorrect.

datanucleus.transactionIsolation

- Default Value: `read-committed`
- Added In: Hive 0.7.0

Default transaction isolation level for identity generation.

datanucleus.cache.level2

- Default Value: `false`
- Added In: Hive 0.7.0

This parameter does nothing.
Warning note: For most installations, Hive should not enable the DataNucleus L2 cache, since this can cause correctness issues. Thus, some people set this parameter to false assuming that this disables the cache – unfortunately, it does not. To actually disable the cache, set **datanucleus.cache.level2.type** to "none".

datanucleus.cache.level2.type

- Default Value: `none` in Hive 0.9 and later; `SOFT` in Hive 0.7 to 0.8.1
- Added In: Hive 0.7.0

NONE = disable the datanucleus level 2 cache, SOFT = soft reference based cache, WEAK = weak reference based cache.
Warning note: For most Hive installations, enabling the datanucleus cache can lead to correctness issues, and is dangerous. This should be left as "none".

datanucleus.identifierFactory

- Default Value: `datanucleus`
- Added In: Hive 0.7.0

Name of the identifier factory to use when generating table/column names etc. 'datanucleus' is used for backward compatibility.

datanucleus.plugin.pluginRegistryBundleCheck

- Default Value: `LOG`
- Added In: Hive 0.7.0

Defines what happens when plugin bundles are found and are duplicated: EXCEPTION, LOG, or NONE.

hive.metastore.warehouse.dir

- Default Value: `/user/hive/warehouse`
- Added In: Hive 0.2.0

Location of default database for the warehouse.

hive.warehouse.subdir.inherit.perms

- Default Value: `false`
- Added In: Hive 0.9.0 with HIVE-2504.
- Removed In: Hive 3.0.0 with HIVE-16392

Set this to true if table directories should inherit the permissions of the warehouse or database directory instead of being created with permissions derived from dfs umask. (This configuration property replaced **hive.files.umask.value** before Hive 0.9.0 was released) (This configuration property was removed in release 3.0.0, more details in Permission Inheritance in Hive)

Behavior of the flag is changed with Hive-0.14.0 in HIVE-6892 and sub-JIRA's. More details in Permission Inheritance in Hive.

hive.metastore.execute.setugi

- Default Value: `false` in Hive 0.8.1 through 0.13.0, `true` starting in Hive 0.14.0
- Added In: Hive 0.8.1 with HIVE-2616, default changed in Hive 0.14.0 with HIVE-6903

In unsecure mode, true will cause the metastore to execute DFS operations using the client's reported user and group permissions. Note that this property must be set on both the client and server sides. Further note that it's best effort. If client sets it to true and server sets it to false, the client setting will be ignored.

hive.metastore.event.listeners

- Default Value: (empty)
- Added In: Hive 0.8.0 with HIVE-2038

List of comma-separated listeners for metastore events.

hive.metastore.partition.inherit.table.properties

- Default Value: (empty)
- Added In: Hive 0.8.1

List of comma-separated keys occurring in table properties which will get inherited to newly created partitions. * implies all the keys will get inherited.

hive.metastore.end.function.listeners

- Default Value: (empty)
- Added In: Hive 0.8.1

List of comma-separated listeners for the end of metastore functions.

hive.metastore.event.expiry.duration

- Default Value: `0`
- Added In: Hive 0.8.0

Duration after which events expire from events table (in seconds).

hive.metastore.event.clean.freq

- Default Value: `0`
- Added In: Hive 0.8.0

Frequency at which timer task runs to purge expired events in metastore(in seconds).

hive.metastore.connect.retries

- Default Value: `3`
- Added In: Hive 0.6.0

Number of retries while opening a connection to metastore.

hive.metastore.client.connect.retry.delay

- Default Value: `1`
- Added In: Hive 0.7.0

Number of seconds for the client to wait between consecutive connection attempts.

hive.metastore.client.socket.timeout

- Default Value: `20` in Hive 0.7 through 0.13.1; `600` in Hive 0.14.0 and later
- Added In: Hive 0.7.0; default changed in Hive 0.14.0 with HIVE-7140

MetaStore Client socket timeout in seconds.

hive.metastore.rawstore.impl

- Default Value: `org.apache.hadoop.hive.metastore.ObjectStore`
- Added In: Hive 0.8.1

Name of the class that implements org.apache.hadoop.hive.metastore.rawstore interface. This class is used to store and retrieval of raw metadata objects such as table, database.

hive.metastore.batch.retrieve.max

- Default Value: `300`
- Added In: Hive 0.8.0

Maximum number of objects (tables/partitions) can be retrieved from metastore in one batch. The higher the number, the less the number of round trips is needed to the Hive metastore server, but it may also cause higher memory requirement at the client side.

hive.metastore.ds.connection.url.hook

- Default Value: (empty)
- Added In: Hive 0.6.0

Name of the hook to use for retriving the JDO connection URL. If empty, the value in javax.jdo.option.ConnectionURL is used.

hive.metastore.ds.retry.attempts

- Default Value: `1`
- Added In: Hive 0.6.0

The number of times to retry a metastore call if there were a connection error.

hive.metastore.ds.retry.interval

- Default Value: `1000`
- Added In: Hive 0.6.0

The number of milliseconds between metastore retry attempts.

hive.metastore.server.min.threads

- Default Value: `200`
- Added In: Hive 0.6.0 with HIVE-1270

Minimum number of worker threads in the Thrift server's pool.

hive.metastore.server.max.threads

- Default Value:
  - Hive 0.x and 1.0.x: `100000`
  - Hive 1.1.0 and later: `1000` (HIVE-8666)
- Added In: Hive 0.6.0 with HIVE-1270

Maximum number of worker threads in the Thrift server's pool.

hive.metastore.server.max.message.size

- Default Value: `100*1024*1024`
- Added In: Hive 1.1.0 (backported to Hive 1.0.2) with HIVE-8680

Maximum message size in bytes a Hive metastore will accept.

hive.metastore.server.tcp.keepalive

- Default Value: `true`
- Added In: Hive 0.6.0

Whether to enable TCP keepalive for the metastore server. Keepalive will prevent accumulation of half-open connections.

hive.metastore.sasl.enabled

- Default Value: `false`
- Added In: Hive 0.7.0

If true, the metastore thrift interface will be secured with SASL. Clients must authenticate with Kerberos.

hive.metastore.kerberos.keytab.file

- Default Value: (empty)
- Added In: Hive 0.7.0

The path to the Kerberos Keytab file containing the metastore thrift server's service principal.

hive.metastore.kerberos.principal

- Default Value: `hive-metastore/_HOST@EXAMPLE.COM`
- Added In: Hive 0.7.0

The service principal for the metastore thrift server. The special string _HOST will be replaced automatically with the correct host name.

Note: This principal is used by the metastore process for authentication with other services (e.g. for HDFS operations).

hive.metastore.client.kerberos.principal

- Default Value: (empty)
- Added In: Hive 2.2.1, 2.4.0 (HIVE-17489)

The client-facing Kerberos service principal for the Hive metastore. If unset, it defaults to the value set for **hive.metastore.kerberos.principal**, for backward compatibility.

Also see **hive.server2.authentication.client.kerberos.principal**.

hive.metastore.cache.pinobjtypes

- Default Value: `Table,StorageDescriptor,SerDeInfo,Partition,Database,Type,FieldSchema,Order`
- Added In: Hive 0.7.0

List of comma-separated metastore object types that should be pinned in the cache.

hive.metastore.authorization.storage.checks

- Default Value: `false`
- Added In: Hive 0.8.0

Should the metastore do authorization checks against the underlying storage for operations like drop-partition (disallow the drop-partition if the user in question doesn't have permissions to delete the corresponding directory on the storage).

hive.metastore.thrift.framed.transport.enabled

- Default Value: `false`
- Added In: Hive 0.10.0 with HIVE-2767

If true, the metastore Thrift interface will use TFramedTransport. When false (default) a standard TTransport is used.

hive.metastore.schema.verification

- Default Value: `false`
- Added In: Hive 0.12.0 with HIVE-3764

Enforce metastore schema version consistency.
True: Verify that version information stored in metastore matches with one from Hive jars. Also disable automatic schema migration attempt (see **datanucleus.autoCreateSchema** and **datanucleus.schema.autoCreateAll**). Users are required to manually migrate schema after Hive upgrade which ensures proper metastore schema migration.
False: Warn if the version information stored in metastore doesn't match with one from Hive jars.

For more information, see Metastore Schema Consistency and Upgrades.

hive.metastore.disallow.incompatible.col.type.changes

- Default Value:
  - Hive 0.x and 1.x: `false`
  - Hive 2.x and later: `true` (HIVE-12320)
- Added In: Hive 0.12.0 with HIVE-4409

If true, ALTER TABLE operations which change the type of a column (say STRING) to an incompatible type (say MAP<STRING, STRING>) are disallowed. RCFile default SerDe (ColumnarSerDe) serializes the values in such a way that the datatypes can be converted from string to any type. The map is also serialized as a string, which can be read as a string as well. However, with any binary serialization, this is not true. Blocking the ALTER TABLE prevents ClassCastException when subsequently trying to access old partitions.

Primitive types like INT, STRING, BIGINT, etc. are compatible with each other and are not blocked.

See HIVE-4409 for more details.

hive.metastore.integral.jdo.pushdown

- Default Value: `false`
- Added In: Hive 0.13.0 with HIVE-6052

Allow JDO query pushdown for integral partition columns in metastore. Off by default. This improves metastore performance for integral columns, especially if there's a large number of partitions. However, it doesn't work correctly with integral values that are not normalized (for example, if they have leading zeroes like 0012). If metastore direct SQL is enabled and works (**hive.metastore.try.direct.sql**), this optimization is also irrelevant.

hive.metastore.try.direct.sql

- Default Value: `true`
- Added In: Hive 0.12.0 with HIVE-4051

Whether the Hive metastore should try to use direct SQL queries instead of the DataNucleus for certain read paths. This can improve metastore performance when fetching many partitions or column statistics by orders of magnitude; however, it is not guaranteed to work on all RDBMS-es and all versions. In case of SQL failures, the metastore will fall back to the DataNucleus, so it's safe even if SQL doesn't work for all queries on your datastore. If all SQL queries fail (for example, your metastore is backed by MongoDB), you might want to disable this to save the try-and-fall-back cost.

This can be configured on a per client basis by using the "set metaconf:hive.metastore.try.direct.sql=<value>" command, starting with Hive 0.14.0 (HIVE-7532).

hive.metastore.try.direct.sql.ddl

- Default Value: `true`
- Added In: Hive 0.13.0 with HIVE-5626

Same as **hive.metastore.try.direct.sql**, for read statements within a transaction that modifies metastore data. Due to non-standard behavior in Postgres, if a direct SQL select query has incorrect syntax or something similar inside a transaction, the entire transaction will fail and fall-back to DataNucleus will not be possible. You should disable the usage of direct SQL inside transactions if that happens in your case.

This can be configured on a per client basis by using the "set metaconf:hive.metastore.try.direct.sql.ddl=<value>" command, starting with Hive 0.14.0 (HIVE-7532).

hive.metastore.orm.retrieveMapNullsAsEmptyStrings

- Default Value: `false`
- Added In: Hive 1.0.0 with HIVE-8485

Thrift does not support nulls in maps, so any nulls present in maps retrieved from object-relational mapping (ORM) must be either pruned or converted to empty strings. Some backing databases such as Oracle persist empty strings as nulls, and therefore will need to have this parameter set to `true` in order to reverse that behavior. For others, the default pruning behavior is correct.

hive.direct.sql.max.query.length

- Default Value: `100`
- Added In: Hive 1.3.0 and 2.1.0 (but not 2.0.x) with HIVE-12349

The maximum size of a query string (in KB) as generated by direct SQL.

hive.direct.sql.max.elements.in.clause

- Default Value: `1000`
- Added In: Hive 1.3.0 and 2.1.0 (but not 2.0.x) with HIVE-12349

The maximum number of values in an IN clause as generated by direct SQL. Once exceeded, it will be broken into multiple OR separated IN clauses.

hive.direct.sql.max.elements.values.clause

- Default Value: `1000`
- Added In: Hive 1.3.0 and 2.1.0 (but not 2.0.x) with HIVE-12349

The maximum number of values in a VALUES clause for an INSERT statement as generated by direct SQL.

hive.metastore.port

- Default Value: `9083`
- Added In: Hive 1.3.0 with HIVE-9365

Hive metastore listener port.

hive.metastore.initial.metadata.count.enabled

- Default Value: `true`
- Added In: Hive 2.1.0 with HIVE-12628

Enable a metadata count at metastore startup for metrics.

hive.metastore.limit.partition.request

- Default value: -1
- Added In: Hive 2.2.0 with HIVE-13884

This limits the number of partitions that can be requested from the Metastore for a given table. A query will not be executed if it attempts to fetch more partitions per table than the limit configured. A value of "-1" means unlimited. This parameter is preferred over **hive.limit.query.max.table.partition** (deprecated; removed in 3.0.0).

hive.metastore.fastpath

- Default Value: false
- Added In: Hive 2.0.0 with HIVE-9453

Used to avoid all of the proxies and object copies in the metastore. Note, if this is set, you MUST use a local metastore (`hive.metastore.uris` must be empty) otherwise undefined and most likely undesired behavior will result.

Hive Metastore Connection Pooling Configuration

The Hive Metastore supports several connection pooling implementations (e.g. hikaricp, bonecp, dbcp). Configuration properties prefixed by 'hikari' or 'dbcp' will be propagated as is to the connectionpool implementation by Hive. Jdbc connection url, username, password and connection pool maximum connections are exceptions which must be configured with their special Hive Metastore configuration properties.

Added in: Hive 3.0.0 with HIVE-17318 and HIVE-17319.

Hive Metastore HBase

Development of an HBase metastore for Hive started in release 2.0.0 (HIVE-9452) but the work has been stopped and the code was removed from Hive in release 3.0.0 (HIVE-17234).

Many more configuration properties were created for the HBase metastore in releases 2.x.x – they are not documented here. For a full list, see the doc note on HIVE-17234.

hive.metastore.hbase.cache.size

(This configuration property should never have been documented here, because it was removed before the initial release by HIVE-9693.)

hive.metastore.hbase.cache.ttl

- Default Value: `600s`
- Added In: Hive 2.0.0 with HIVE-9453
- Removed In: Hive 3.0.0 with HIVE-17234

Number of seconds for stats items to live in the cache.

hive.metastore.hbase.file.metadata.threads

- Default Value: `1`
- Added In: Hive 2.1.0 with HIVE-12075

Number of threads to use to read file metadata in background to cache it.

HiveServer2

HiveServer2 was added in Hive 0.11.0 with HIVE-2935. For more information see HiveServer2 Overview, Setting Up HiveServer2, and HiveServer2 Clients.

Besides the configuration properties listed in this section, some HiveServer2 properties are listed in other sections:

- **hive.server2.builtin.udf.whitelist**
- **hive.server2.builtin.udf.blacklist**
- **hive.server2.metrics.enabled**

hive.server2.thrift.port

- Default Value: `10000`
- Added In: Hive 0.11.0 with HIVE-2935

Port number of HiveServer2 Thrift interface. Can be overridden by setting $HIVE_SERVER2_THRIFT_PORT.

hive.server2.thrift.bind.host

- Default Value: `localhost`
- Added In: Hive 0.11.0 with HIVE-2935

Bind host on which to run the HiveServer2 Thrift interface. Can be overridden by setting $HIVE_SERVER2_THRIFT_BIND_HOST.

hive.server2.thrift.min.worker.threads

- Default Value: `5`
- Added In: Hive 0.11.0 with HIVE-2935

Minimum number of Thrift worker threads.

hive.server2.thrift.max.worker.threads

- Default Value: `100` in Hive 0.11.0, `500` in Hive 0.12.0 and later

- Added In: Hive 0.11.0 with HIVE-2935, default value changed in HIVE 0.12.0 with HIVE-4617

Maximum number of Thrift worker threads.

hive.server2.thrift.worker.keepalive.time

- Default Value: `60`
- Added in: Hive 0.14.0 with HIVE-7353

Keepalive time (in seconds) for an idle worker thread. When number of workers > min workers, excess threads are killed after this time interval.

hive.server2.thrift.max.message.size

- Default Value: `100*1024*1024`
- Added in: Hive 1.1.0 (backported to Hive 1.0.2) with HIVE-8680

Maximum message size in bytes a HiveServer2 server will accept.

hive.server2.authentication

- Default Value: `NONE`
- Added In: Hive 0.11.0 with HIVE-2935

Client authentication types.

> NONE: no authentication check – plain SASL transport
> LDAP: LDAP/AD based authentication
> KERBEROS: Kerberos/GSSAPI authentication
> CUSTOM: Custom authentication provider (use with property **hive.server2.custom.authentication.class**)
> PAM: Pluggable authentication module (added in Hive 0.13.0 with HIVE-6466)
> NOSASL: Raw transport (added in Hive 0.13.0)

hive.server2.authentication.kerberos.keytab

- Default Value: (empty)
- Added In: Hive 0.11.0 with HIVE-2935

Kerberos keytab file for server principal.

hive.server2.authentication.kerberos.principal

- Default Value: (empty)
- Added In: Hive 0.11.0 with HIVE-2935

Kerberos server principal.

hive.server2.authentication.client.kerberos.principal

- Default Value: (empty)
- Added In: Hive 2.1.1, 2.4.0 with HIVE-17489

Kerberos server principal used by the HA HiveServer2. Also see **hive.metastore.client.kerberos.principal**.

hive.server2.custom.authentication.class

- Default Value: (empty)
- Added In: Hive 0.11.0 with HIVE-2935

Custom authentication class. Used when property **hive.server2.authentication** is set to 'CUSTOM'. Provided class must be a proper implementation of the interface org.apache.hive.service.auth.PasswdAuthenticationProvider. HiveServer2 will call its Authenticate(user, passed) method to authenticate requests. The implementation may optionally extend Hadoop's org.apache.hadoop.conf.Configured class to grab Hive's Configuration object.

hive.server2.enable.doAs

- Default Value: `true`
- Added In: Hive 0.11.0 with HIVE-2935 and HIVE-4356

Setting this property to true will have HiveServer2 execute Hive operations as the user making the calls to it.

hive.server2.authentication.ldap.url

- Default Value: (empty)
- Added In: Hive 0.11.0 with HIVE-2935

LDAP connection URL(s), value could be a SPACE separated list of URLs to multiple LDAP servers for resiliency. URLs are tried in the order specified until the connection is successful.

hive.server2.authentication.ldap.baseDN

- Default Value: (empty)
- Added In: Hive 0.11.0 with HIVE-2935

LDAP base DN (distinguished name).

hive.server2.authentication.ldap.guidKey

- Default Value: `uid`
- Added In: Hive 2.1.0 with HIVE-13295

This property is to indicate what prefix to use when building the bindDN for LDAP connection (when using just baseDN). So bindDN will be "<guidKey>=<user/group>,<baseDN>". If userDNPattern and/or groupDNPattern is used in the configuration, the guidKey is not needed. Primarily required when just baseDN is being used.

hive.server2.authentication.ldap.Domain

- Default Value: (empty)
- Added In: Hive 0.12.0 with HIVE-4707

LDAP domain.

hive.server2.authentication.ldap.groupDNPattern

- Default Value: (empty)

- Added In: Hive 1.3 with HIVE-7193

A COLON-separated list of string patterns to represent the base DNs for LDAP Groups. Use "%s" where the actual group name is to be plugged in. See Group Membership for details.

Example of one string pattern: *uid=%s,OU=Groups,DC=apache,DC=org*

hive.server2.authentication.ldap.groupFilter

- Default Value: (empty)
- Added In: Hive 1.3 with HIVE-7193

A COMMA-separated list of group names that the users should belong to (at least one of the groups) for authentication to succeed. See Group Membership for details.

hive.server2.authentication.ldap.groupMembershipKey

- Default Value: `member`
- Added In: Hive 2.1.0 with HIVE-13295
- Extended In: Hive 2.1.1 with HIVE-14513

LDAP attribute name on the group object that contains the list of distinguished names for the user, group, and contact objects that are members of the group. For example: member, uniqueMember, or memberUid.

This property is used in LDAP search queries when finding LDAP group names that a particular user belongs to. The value of the LDAP attribute, indicated by this property, should be a full DN for the user or the short username or userid. For example, a group entry for "fooGroup" containing "member : uid=fooUser,ou=Users,dc=domain,dc=com" will help determine that "fooUser" belongs to LDAP group "fooGroup".

See Group Membership for a detailed example.

This property can also be used to find the users if a custom-configured LDAP query returns a group instead of a user (as of Hive 2.1.1). For details, see Support for Groups in Custom LDAP Query.

hive.server2.authentication.ldap.userMembershipKey

- Default Value: `null`
- Added In: Hive 2.2.0 with HIVE-15076

LDAP attribute name on the user object that contains groups of which the user is a direct member, except for the primary group, which is represented by the primaryGroupId. For example: memberOf.

hive.server2.authentication.ldap.groupClassKey

- Default Value: `groupOfNames`
- Added In: Hive 1.3 with HIVE-13295

This property is used in LDAP search queries for finding LDAP group names a user belongs to. The value of this property is used to construct LDAP group search query and is used to indicate what a group's objectClass is. Every LDAP group has certain objectClass. For example: group, groupOfNames, groupOfUniqueNames etc.

See Group Membership for a detailed example.

hive.server2.authentication.ldap.userDNPattern

- Default Value: (empty)
- Added In: Hive 1.3 with HIVE-7193

A COLON-separated list of string patterns to represent the base DNs for LDAP Users. Use "%s" where the actual username is to be plugged in. See User Search List for details.

Example of one string pattern: *uid=%s,OU=Users,DC=apache,DC=org*

hive.server2.authentication.ldap.userFilter

- Default Value: (empty)
- Added In: Hive 1.3 with HIVE-7193

A COMMA-separated list of usernames for whom authentication will succeed if the user is found in LDAP. See User Search List for details.

hive.server2.authentication.ldap.customLDAPQuery

- Default Value: (empty)
- Added In: Hive 1.3 with HIVE-7193

A user-specified custom LDAP query that will be used to grant/deny an authentication request. If the user is part of the query's result set, authentication succeeds. See Custom Query String for details.

hive.server2.global.init.file.location

- Default Value: $HIVE_CONF_DIR (typically <hive_root>/conf)
- Added in Hive 0.14.0 with HIVE-5160, HIVE-7497, and HIVE-8138

Either the location of a HiveServer2 global init file or a directory containing a .hiverc file. If the property is set, the value must be a valid path to an init file or directory where the init file is located.

hive.server2.transport.mode

- Default Value: `binary`
- Added In: Hive 0.12.0
- Deprecated In: Hive 0.14.0 with HIVE-6972 (see Connection URL When HiveServer2 Is Running in HTTP Mode) but only for clients. This setting is still in use and not deprecated for the HiveServer2 itself.

Server transport mode. Value can be "binary" or "http".

hive.server2.thrift.http.port

- Default Value: `10001`
- Added In: Hive 0.12.0

Port number when in HTTP mode.

hive.server2.thrift.http.path

- Default Value: `cliservice`
- Added In: Hive 0.12.0
- Deprecated In: Hive 0.14.0 with HIVE-6972 (see Connection URL When HiveServer2 Is Running in HTTP Mode)

Path component of URL endpoint when in HTTP mode.

hive.server2.thrift.http.min.worker.threads

- Default Value: `5`
- Added In: Hive 0.12.0

Minimum number of worker threads when in HTTP mode.

hive.server2.thrift.http.max.worker.threads

- Default Value: `500`
- Added In: Hive 0.12.0

Maximum number of worker threads when in HTTP mode.

hive.server2.thrift.http.max.idle.time

- Default Value: 1800s (ie, 1800 seconds)
- Added In: Hive 0.14.0 in HIVE-7169

Maximum idle time for a connection on the server when in HTTP mode.

hive.server2.thrift.http.worker.keepalive.time

- Default Value: 60
- Added In: Hive 0.14.0 in HIVE-7353

Keepalive time (in seconds) for an idle http worker thread. When number of workers > min workers, excess threads are killed after this time interval.

hive.server2.thrift.sasl.qop

- Default Value: `auth`
- Added In: Hive 0.12.0

Sasl QOP value; set it to one of the following values to enable higher levels of protection for HiveServer2 communication with clients.

> "auth" – authentication only (default)
> "auth-int" – authentication plus integrity protection
> "auth-conf" – authentication plus integrity and confidentiality protection

Note that hadoop.rpc.protection being set to a higher level than HiveServer2 does not make sense in most situations. HiveServer2 ignores hadoop.rpc.protection in favor of hive.server2.thrift.sasl.qop.

This is applicable only if HiveServer2 is configured to use Kerberos authentication.

hive.server2.async.exec.threads

- Default Value: `50` in Hive 0.12.0, `100` in Hive 0.13.0 and later
- Added In: Hive 0.12.0 with HIVE-4617, default value changed in Hive 0.13.0 with HIVE-5229

Number of threads in the async thread pool for HiveServer2.

hive.server2.async.exec.shutdown.timeout

- Default Value: `10`
- Added In: Hive 0.12.0 with HIVE-4617

Time (in seconds) for which HiveServer2 shutdown will wait for async threads to terminate.

hive.server2.table.type.mapping

- Default Value: `CLASSIC`
- Added In: Hive 0.12.0 with HIVE-4573 and HIVE-4998

This setting reflects how HiveServer2 will report the table types for JDBC and other client implementations that retrieve the available tables and supported table types.

> HIVE: Exposes Hive's native table types like MANAGED_TABLE, EXTERNAL_TABLE, VIRTUAL_VIEW
> CLASSIC: More generic types like TABLE and VIEW

hive.server2.session.hook

- Default Value: (empty)
- Added In: Hive 0.12.0 with HIVE-4588

Session-level hook for HiveServer2.

hive.server2.max.start.attempts

- Default Value: `30`
- Added In: Hive 0.13.0 with HIVE-5794

The number of times HiveServer2 will attempt to start before exiting, sleeping 60 seconds between retries. The default of 30 will keep trying for 30 minutes.

hive.server2.async.exec.wait.queue.size

- Default Value: `100`
- Added In: Hive 0.13.0 with HIVE-5229

Size of the wait queue for async thread pool in HiveServer2. After hitting this limit, the async thread pool will reject new requests.

hive.server2.async.exec.keepalive.time

- Default Value: `10`
- Added In: Hive 0.13.0 with HIVE-5229

Time (in seconds) that an idle HiveServer2 async thread (from the thread pool) will wait for a new task to arrive before terminating.

hive.server2.long.polling.timeout

- Default Value: `5000L`
- Added In: Hive 0.13.0 with HIVE-5217

Time in milliseconds that HiveServer2 will wait, before responding to asynchronous calls that use long polling.

hive.server2.allow.user.substitution

- Default Value: `true`
- Added In: Hive 0.13.0

Allow alternate user to be specified as part of HiveServer2 open connection request.

hive.server2.authentication.spnego.keytab

- Default Value: (empty)
- Added In: Hive 0.13.0

Keytab file for SPNEGO principal, optional. A typical value would look like `/etc/security/keytabs/spnego.service.keytab`. This keytab would be used by HiveServer2 when Kerberos security is enabled and HTTP transport mode is used. This needs to be set only if SPNEGO is to be used in authentication.

SPNEGO authentication would be honored only if valid **hive.server2.authentication.spnego.principal** and **hive.server2.authentication.spnego.keytab** are specified.

hive.server2.authentication.spnego.principal

- Default Value: (empty)
- Added In: Hive 0.13.0

SPNEGO service principal, optional. A typical value would look like `HTTP/_HOST@EXAMPLE.COM`. The SPNEGO service principal would be used by HiveServer2 when Kerberos security is enabled and HTTP transport mode is used. This needs to be set only if SPNEGO is to be used in authentication.

hive.server2.authentication.pam.services

- Default Value: (empty)
- Added In: Hive 0.13.0 with HIVE-6466

List of the underlying PAM services that should be used when **hive.server2.authentication** type is PAM. A file with the same name must exist in /etc/pam.d.

hive.server2.use.SSL

- Default Value: `false`
- Added In: Hive 0.13.0 with HIVE-5351

Set this to true for using SSL encryption in HiveServer2.

hive.server2.keystore.path

- Default Value: (empty)
- Added In: Hive 0.13.0 with HIVE-5351

SSL certificate keystore location.

hive.server2.keystore.password

- Default Value: (empty)
- Added In: Hive 0.13.0 with HIVE-5351

SSL certificate keystore password.

hive.server2.tez.default.queues

- Default Value: (empty)
- Added In: Hive 0.13.0 with HIVE-6325

A list of comma separated values corresponding to YARN queues of the same name. When HiveServer2 is launched in Tez mode, this configuration needs to be set for multiple Tez sessions to run in parallel on the cluster.

hive.server2.tez.sessions.per.default.queue

- Default Value: `1`
- Added In: Hive 0.13.0 with HIVE-6325

A positive integer that determines the number of Tez sessions that should be launched on each of the queues specified by **hive.server2.tez.default.queues**. Determines the parallelism on each queue.

hive.server2.tez.initialize.default.sessions

- Default Value: `false`
- Added In: Hive 0.13.0 with HIVE-6325

This flag is used in HiveServer 2 to enable a user to use HiveServer 2 without turning on Tez for HiveServer 2. The user could potentially want to run queries over Tez without the pool of sessions.

hive.server2.session.check.interval

- Default Value:
  - Hive 0.x, 1.0.x, 1.1.x, 1.2.0: `0ms`
  - Hive 1.2.1+, 1.3+, 2.x+: `6h` (HIVE-9842)
- Added In: Hive 0.14.0 with HIVE-5799

The check interval for session/operation timeout, which can be disabled by setting to zero or negative value.

hive.server2.idle.session.timeout

- Default Value:
  - Hive 0.x, 1.0.x, 1.1.x, 1.2.0: `0ms`
  - Hive 1.2.1+, 1.3+, 2.x+: `7d` (HIVE-9842)
- Added In: Hive 0.14.0 with HIVE-5799

With **hive.server2.session.check.interval** set to a positive time value, session will be closed when it's not accessed for this duration of time, which can be disabled by setting to zero or negative value.

hive.server2.idle.operation.timeout

- Default Value: 0ms

- Added In: Hive 0.14.0 with HIVE-5799

With **hive.server2.session.check.interval** set to a positive time value, operation will be closed when it's not accessed for this duration of time, which can be disabled by setting to zero value.

> With positive value, it's checked for operations in terminal state only (FINISHED, CANCELED, CLOSED, ERROR).
> With negative value, it's checked for all of the operations regardless of state.

hive.server2.logging.operation.enabled

- Default Value: `true`
- Added In: Hive 0.14.0 with HIVE-4629 and HIVE-8785

When true, HiveServer2 will save operation logs and make them available for clients.

hive.server2.logging.operation.log.location

- Default Value: `${java.io.tmpdir}/${user.name}/operation_logs`
- Added In: Hive 0.14.0 with HIVE-4629

Top level directory where operation logs are stored if logging functionality is enabled.

hive.server2.logging.operation.verbose

- Default Value: `false`
- Added In: Hive 0.14.0 with HIVE-8785
- Removed In: Hive 1.2.0 with HIVE-10119

When `true`, HiveServer2 operation logs available for clients will be verbose. Replaced in Hive 1.2.0 by **hive.server2.logging.operation.level**.

hive.server2.logging.operation.level

- Default Value: EXECUTION
- Added In: Hive 1.2.0 with HIVE-10119

HiveServer2 operation logging mode available to clients to be set at session level.

For this to work, **hive.server2.logging.operation.enabled** should be set to true. The allowed values are:

- NONE: Ignore any logging.
- EXECUTION: Log completion of tasks.
- PERFORMANCE: Execution + Performance logs.
- VERBOSE: All logs.

hive.server2.thrift.http.cookie.auth.enabled

- Default Value: `true`
- Added In: Hive 1.2.0 with HIVE-9710

When true, HiveServer2 in HTTP transport mode will use cookie based authentication mechanism.

hive.server2.thrift.http.cookie.max.age

- Default Value: `86400s` (1 day)
- Added In: Hive 1.2.0 with HIVE-9710

Maximum age in seconds for server side cookie used by HiveServer2 in HTTP mode.

hive.server2.thrift.http.cookie.path

- Default Value: (empty)
- Added In: Hive 1.2.0 with HIVE-9710

Path for the HiveServer2 generated cookies.

hive.server2.thrift.http.cookie.domain

- Default Value: (empty)
- Added In: Hive 1.2.0 with HIVE-9710

Domain for the HiveServer2 generated cookies.

hive.server2.thrift.http.cookie.is.secure

- Default Value: `true`
- Added In: Hive 1.2.0 with HIVE-9710

Secure attribute of the HiveServer2 generated cookie.

hive.server2.thrift.http.cookie.is.httponly

- Default Value: `true`
- Added In: Hive 1.2.0 with HIVE-9710

HttpOnly attribute of the HiveServer2 generated cookie.

hive.server2.close.session.on.disconnect

- Default Value: `true`
- Added In: Hive 2.1.0 with HIVE-13415

Session will be closed when connection is closed. Set this to false to have session outlive its parent connection.

hive.server2.xsrf.filter.enabled

- Default Value: `false`
- Added In: Hive 2.2.0 with HIVE-13853

If enabled, HiveServer2 will block any requests made to it over HTTP if an X-XSRF-HEADER header is not present.

hive.server2.job.credential.provider.path

- Default Value: (empty)
- Added In: Hive 2.2.0 with HIVE-14822

This configuration property enables the user to provide a comma-separated list of URLs which provide the type and location of Hadoop credential providers. These credential providers are used by HiveServer2 for providing job-specific credentials launched using MR or Spark execution engines. This functionality has not been tested against Tez.

hive.server2.in.place.progress

- Default Value: `true`
- Added In: Hive 2.2.0 with HIVE-15473

Allows HiveServer2 to send progress bar update information. This is currently available only if the execution engine is **tez.**

hive.hadoop.classpath

- Default Value: (empty)
- Added In: Hive 0.14.0 with HIVE-8340

For the Windows operating system, Hive needs to pass the HIVE_HADOOP_CLASSPATH Java parameter while starting HiveServer2 using "`-hiveconf hive.hadoop.classpath=%HIVE_LIB%`". Users can set this parameter in hiveserver2.xml.

HiveServer2 Web UI

A web interface for HiveServer2 is introduced in release 2.0.0 (see Web UI for HiveServer2).

hive.server2.webui.host

- Default Value: `0.0.0.0`
- Added In: Hive 2.0.0 with HIVE-12338

The host address the HiveServer2 Web UI will listen on. The Web UI can be used to access the HiveServer2 configuration, local logs, and metrics. It can also be used to check some information about active sessions and queries being executed.

hive.server2.webui.port

- Default Value: `10002`
- Added In: Hive 2.0.0 with HIVE-12338 and HIVE-12711

The port the HiveServer2 Web UI will listen on. Set to 0 or a negative number to disable the HiveServer2 Web UI feature.

hive.server2.webui.max.threads

- Default Value: `50`
- Added In: Hive 2.0.0 with HIVE-12338

The maximum number of HiveServer2 Web UI threads.

hive.server2.webui.max.historic.queries

- Default Value: `25`
- Added In: Hive 2.1.0 with HIVE-12550

The maximum number of past queries to show in HiveServer2 Web UI.

hive.server2.webui.use.ssl

- Default Value: `false`
- Added In: Hive 2.0.0 with HIVE-12471 and HIVE-12485

Set this to true for using SSL encryption for HiveServer2 WebUI.

hive.server2.webui.keystore.path

- Default Value: (empty)
- Added In: Hive 2.0.0 with HIVE-12471

SSL certificate keystore location for HiveServer2 WebUI.

hive.server2.webui.keystore.password

- Default Value: (empty)
- Added In: Hive 2.0.0 with HIVE-12471

SSL certificate keystore password for HiveServer2 WebUI.

hive.server2.webui.use.spnego

- Default Value: `false`
- Added In: Hive 2.0.0 with HIVE-12485

SSL certificate keystore password for HiveServer2 WebUI.

hive.server2.webui.spnego.keytab

- Default Value: (empty)
- Added In: Hive 2.0.0 with HIVE-12485

The path to the Kerberos Keytab file containing the HiveServer2 WebUI SPNEGO service principal.

hive.server2.webui.spnego.principal

- Default Value: `HTTP/_HOST@EXAMPLE.COM`
- Added In: Hive 2.0.0 with HIVE-12485

The HiveServer2 WebUI SPNEGO service principal. The special string _HOST will be replaced automatically with the value of **hive.server2.webui.host** or the correct host name.

hive.server2.webui.explain.output

- Default Value: `false`
- Added in: Hive 3.1.0 with HIVE-18469

The EXPLAIN EXTENDED output for the query will be shown in the WebUI / Drilldown / Query Plan tab when this configuration property is set to true.

Prior to Hive 3.1.0, you can use **hive.log.explain.output** instead of this configuration property.

Spark

Apache Spark was added in Hive 1.1.0 (HIVE-7292 and the merge-to-trunk JIRA's HIVE-9257, 9352, 9448). For information see the design document Hive on Spark and Hive on Spark: Getting Started.

To configure Hive execution to Spark, set the following property to "`spark`":

- **hive.execution.engine**

hive.spark.job.monitor.timeout

- Default Value: `60` seconds
- Added In: Hive 1.1.0 with HIVE-9337

Timeout for job monitor to get Spark job state.

hive.spark.dynamic.partition.pruning

- Default Value: `false`
- Added In: Hive 1.3.0 with HIVE-9152

When true, this turns on dynamic partition pruning for the Spark engine, so that joins on partition keys will be processed by writing to a temporary HDFS file, and read later for removing unnecessary partitions.

hive.spark.dynamic.partition.pruning.max.data.size

- Default Value: `100MB`
- Added In: Hive 1.3.0 with HIVE-9152

The maximum data size for the dimension table that generates partition pruning information. If reaches this limit, the optimization will be turned off.

hive.spark.exec.inplace.progress

- Default Value: `true`
- Added In: Hive 2.2.0 with HIVE-15039

Updates Spark job execution progress in-place in the terminal.

hive.spark.use.file.size.for.mapjoin

- Default Value: `false`
- Added In: Hive 2.3.0 with HIVE-15489
- Removed In: Hive 3.0.0 with HIVE-16336, replaced by **hive.spark.use.ts.stats.for.mapjoin**

If this is set to true, mapjoin optimization in Hive/Spark will use source file sizes associated with the TableScan operator on the root of the operator tree, instead of using operator statistics.

hive.spark.use.ts.stats.for.mapjoin

- Default Value: `false`
- Added In: Hive 3.0.0 with HIVE-16336, replaces **hive.spark.use.file.size.for.mapjoin**

If this is set to true, mapjoin optimization in Hive/Spark will use statistics from TableScan operators at the root of the operator tree, instead of parent ReduceSink operators of the Join operator.

hive.spark.explain.user

- Default Value: `false`
- Added In: Hive 3.0.0 with HIVE-11133

Whether to show explain result at user level for Hive-on-Spark queries. When enabled, will log EXPLAIN output for the query at user level.

hive.prewarm.spark.timeout

- Default Value: 5000ms
- Added In: Hive 3.0.0 with HIVE-17362

Time to wait to finish prewarming Spark executors when **hive.prewarm.enabled** is true.

Note: These configuration properties for Hive on Spark are documented in the Tez section because they can also affect Tez:
- **hive.prewarm.enabled**
- **hive.prewarm.numcontainers**

hive.spark.optimize.shuffle.serde

- Default Value: `false`
- Added In: Hive 3.0.0 with HIVE-15104

If this is set to true, Hive on Spark will register custom serializers for data types in shuffle. This should result in less shuffled data.

Remote Spark Driver

The remote Spark driver is the application launched in the Spark cluster, that submits the actual Spark job. It was introduced in HIVE-8528. It is a long-lived application initialized upon the first query of the current user, running until the user's session is closed. The following properties control the remote communication between the remote Spark driver and the Hive client that spawns it.

hive.spark.client.future.timeout

- Default Value: `60` seconds
- Added In: Hive 1.1.0 with HIVE-9337

Timeout for requests from Hive client to remote Spark driver.

hive.spark.client.connect.timeout

- Default Value: `1000` miliseconds
- Added In: Hive 1.1.0 with HIVE-9337

Timeout for remote Spark driver in connecting back to Hive client.

hive.spark.client.server.connect.timeout

- Default Value: `90000` miliseconds
- Added In: Hive 1.1.0 with HIVE-9337, default changed in same release with HIVE-9519

Timeout for handshake between Hive client and remote Spark driver. Checked by both processes.

hive.spark.client.secret.bits

- Default Value: `256`
- Added In: Hive 1.1.0 with HIVE-9337

Number of bits of randomness in the generated secret for communication between Hive client and remote Spark driver. Rounded down to nearest multiple of 8.

hive.spark.client.rpc.server.address

- Default Value: hive.spark.client.rpc.server.address, localhost if unavailable.
- Added In: Hive 2.1.0 with HIVE-12568

The server address of HiverServer2 host to be used for communication between Hive client and remote Spark driver.

hive.spark.client.rpc.threads

- Default Value: `8`
- Added In: Hive 1.1.0 with HIVE-9337

Maximum number of threads for remote Spark driver's RPC event loop.

hive.spark.client.rpc.max.size

- Default Value: `52,428,800` (50 * 1024 * 1024, or 50 MB)
- Added In: Hive 1.1.0 with HIVE-9337

Maximum message size in bytes for communication between Hive client and remote Spark driver. Default is 50 MB.

hive.spark.client.channel.log.level

- Default Value: (empty)
- Added In: Hive 1.1.0 with HIVE-9337

Channel logging level for remote Spark driver. One of DEBUG, ERROR, INFO, TRACE, WARN. If unset, TRACE is chosen.

Tez

Apache Tez was added in Hive 0.13.0 (HIVE-4660 and HIVE-6098). For information see the design document Hive on Tez, especially the Installation and Configuration section.

Besides the configuration properties listed in this section, some properties in other sections are also related to Tez:

- **hive.execution.engine**
- **hive.mapjoin.optimized.hashtable**
- **hive.mapjoin.optimized.hashtable.wbsize**
- **hive.server2.tez.default.queues**
- **hive.server2.tez.sessions.per.default.queue**
- **hive.server2.tez.initialize.default.sessions**
- **hive.stats.max.variable.length**
- **hive.stats.list.num.entries**
- **hive.stats.map.num.entries**
- **hive.stats.map.parallelism** (Hive 0.13 only; removed in Hive 0.14)
- **hive.stats.join.factor**
- **hive.stats.deserialization.factor**
- **hive.tez.dynamic.semijoin.reduction**
- **hive.tez.min.bloom.filter.entries**
- **hive.tez.max.bloom.filter.entries**
- **hive.tez.bloom.filter.factor**
- **hive.tez.bigtable.minsize.semijoin.reduction**
- **hive.explain.user**

hive.jar.directory

- Default Value: `null`
- Added In: Hive 0.13.0 with HIVE-5003 and HIVE-6098, default changed in HIVE-6636

This is the location that Hive in Tez mode will look for to find a site-wide installed Hive instance. See **hive.user.install.directory** for the default behavior.

hive.user.install.directory

- Default Value: `hdfs:///user/`
- Added In: Hive 0.13.0 with HIVE-5003 and HIVE-6098

If Hive (in Tez mode only) cannot find a usable Hive jar in **hive.jar.directory**, it will upload the Hive jar to <**hive.user.install.directory**>/<*user_name*> and use it to run queries.

hive.compute.splits.in.am

- Default Value: `true`
- Added In: Hive 0.13.0 with HIVE-5522 and HIVE-6098

Whether to generate the splits locally or in the ApplicationMaster (Tez only).

hive.rpc.query.plan

- Default Value: `false`
- Added In: Hive 0.13.0 with HIVE-5522 and HIVE-6098

Whether to send the query plan via local resource or RPC.

hive.prewarm.enabled

- Default Value: `false`
- Added In: Hive 0.13.0 with HIVE-6391 and HIVE-6360

Enables container prewarm for Tez (0.13.0 to 1.2.x) or Tez/Spark (1.3.0+). This is for Hadoop 2 only.

hive.prewarm.numcontainers

- Default Value: `10`
- Added In: Hive 0.13.0 with HIVE-6391 and HIVE-6360

Controls the number of containers to prewarm for Tez (0.13.0 to 1.2.x) or Tez/Spark (1.3.0+). This is for Hadoop 2 only.

hive.merge.tezfiles

- Default Value: `false`
- Added In: Hive 0.13.0 with HIVE-6498 and HIVE-6360

Merge small files at the end of a Tez DAG.

hive.tez.input.format

- Default Value: `org.apache.hadoop.hive.ql.io.HiveInputFormat`
- Added In: Hive 0.13.0 with HIVE-6498 and HIVE-6360

The default input format for Tez. Tez groups splits in the AM (ApplicationMaster).

hive.tez.input.generate.consistent.splits

- Default Value: `true`
- Added In: Hive 2.1.0 with HIVE-9850, HIVE-10104 and HIVE-12078

Whether to generate consistent split locations when generating splits in the AM.

Setting to false randomizes the location and order of splits depending on how threads generate.

Relates to LLAP.

hive.tez.container.size

- Default Value: `-1`
- Added In: Hive 0.13.0 with HIVE-6498 and HIVE-6360

By default Tez will spawn containers of the size of a mapper. This can be used to overwrite the default.

hive.tez.java.opts

- Default Value: (empty)
- Added In: Hive 0.13.0 with HIVE-6498 and HIVE-6360

By default Tez will use the Java options from map tasks. This can be used to overwrite the default.

hive.convert.join.bucket.mapjoin.tez

- Default Value: `false`
- Added In: Hive 0.13.0 with HIVE-6447

Whether joins can be automatically converted to bucket map joins in Hive when Tez is used as the execution engine (**hive.execution.engine** is set to "`tez`").

hive.tez.log.level

- Default Value: `INFO`
- Added In: Hive 0.13.0 with HIVE-6743

The log level to use for tasks executing as part of the DAG. Used only if **hive.tez.java.opts** is used to configure Java options.

hive.localize.resource.wait.interval

- Default Value: `5000`
- Added In: Hive 0.13.0 with HIVE-6782

Time in milliseconds to wait for another thread to localize the same resource for Hive-Tez.

hive.localize.resource.num.wait.attempts

- Default Value: `5`
- Added In: Hive 0.13.0 with HIVE-6782

The number of attempts waiting for localizing a resource in Hive-Tez.

hive.tez.smb.number.waves

- Default Value: `0.5`
- Added In: Hive 0.14.0 with HIVE-8409

The number of waves in which to run the SMB (sort-merge-bucket) join. Account for cluster being occupied. Ideally should be 1 wave.

hive.tez.cpu.vcores

- Default Value: `-1`
- Added In: Hive 0.14.0 with HIVE-8452

By default Tez will ask for however many CPUs MapReduce is configured to use per container. This can be used to overwrite the default.

hive.tez.auto.reducer.parallelism

- Default Value: `false`
- Added In: Hive 0.14.0 with HIVE-7158

Turn on Tez' auto reducer parallelism feature. When enabled, Hive will still estimate data sizes and set parallelism estimates. Tez will sample source vertices' output sizes and adjust the estimates at runtime as necessary.

hive.tez.max.partition.factor

- Default Value: `2`
- Added In: Hive 0.14.0 with HIVE-7158

When auto reducer parallelism is enabled this factor will be used to over-partition data in shuffle edges.

hive.tez.min.partition.factor

- Default Value: `0.25`
- Added In: Hive 0.14.0 with HIVE-7158

When auto reducer parallelism is enabled this factor will be used to put a lower limit to the number of reducers that Tez specifies.

hive.tez.exec.print.summary

- Default Value: `false`
- Added In: Hive 0.14.0 with HIVE-8495

If true, displays breakdown of execution steps for every query executed on Hive CLI or Beeline client.

hive.tez.exec.inplace.progress

- Default Value: `true`
- Added In: Hive 0.14.0 with HIVE-8495

Updates Tez job execution progress in-place in the terminal when Hive CLI is used.

LLAP

Live Long and Process (LLAP) functionality was added in Hive 2.0 (HIVE-7926 and associated tasks). For details see LLAP in Hive.

LLAP adds the following configuration properties.

hive.llap.execution.mode

- Default Value: `none`
- Possible Values:
  - `none: not tried`
  - `map: only map operators are considered for llap`
  - `all: every operator is tried; but falls back to no-llap in case of problems`
  - `only: same as "all" but stops with an exception if execution is not possible` (as of 2.2.0 with HIVE-15135)
  - auto: conversion is controlled by hive
- Added In: Hive 2.0.0 with HIVE-9635

Chooses whether query fragments will run in a container or in LLAP. When set to "`all`" everything runs in LLAP if possible; "`only`" is like "`all`" but disables fallback to containers, so that the query fails if it cannot run in LLAP.

hive.server2.llap.concurrent.queries

- Default Value: -1
- Added In: Hive 2.0.0 with HIVE-10647

The number of queries allowed in parallel via llap. Negative number implies 'infinite'.

LLAP Client

hive.llap.client.consistent.splits

- Default Value: false
- Added In: Hive 2.0.0 with HIVE-12470

Whether to setup split locations to match nodes on which LLAP daemons are running, instead of using the locations provided by the split itself.

LLAP Web Services

hive.llap.daemon.web.port

- Default Value: 15002
- Added In: Hive 2.0.0 with HIVE-11358

LLAP daemon web UI port.

hive.llap.daemon.web.ssl

- Default Value: false
- Added In: Hive 2.0.0 with HIVE-11358

Whether LLAP daemon web UI should use SSL

hive.llap.auto.auth

- Default Value: true
- Added In: Hive 2.0.0 with ..

Whether or not to set Hadoop configs to enable auth in LLAP web app.

hive.llap.daemon.service.principal

- Default Value: (empty)
- Added In: Hive 2.0.0 with HIVE-12341

The name of the LLAP daemon's service principal.

hive.llap.daemon.service.hosts

- Default Value: null
- Added In: Hive 2.0.0 with HIVE-11358 and HIVE-12470

Explicitly specified hosts to use for LLAP scheduling. Useful for testing. By default, YARN registry is used.

hive.llap.daemon.task.preemption.metrics.intervals

- Default Value:30,60,300
- Added In: Hive 2.1.0 with HIVE-13536

Comma-delimited set of integers denoting the desired rollover intervals (in seconds) for percentile latency metrics.
Used by LLAP daemon task scheduler metrics for time taken to kill task (due to pre-emption) and useful time wasted by the task that is about to be preempted.

LLAP Cache

hive.llap.object.cache.enabled

- Default Value: true
- Added In: Hive 2.0.0 with HIVE-9849

Cache objects (plans, hashtables, etc) in LLAP

hive.llap.io.use.lrfu

- Default Value: false
- Added In: Hive 2.0.0 with ..

Whether ORC low-level cache should use Least Frequently / Frequently Used (LRFU) cache policy instead of default First-In-First-Out (FIFO).

hive.llap.io.lrfu.lambda

- Default Value: 0.01f
- Possible Values: Between 0 and 1
- Added In: Hive 2.0.0 with ..

Lambda for ORC low-level cache LRFU cache policy. Must be in [0, 1].

0 makes LRFU behave like LFU, 1 makes it behave like LRU, values in between balance accordingly.

LLAP I/O

hive.llap.io.enabled

- Default Value: null
- Added In: Hive 2.0.0 with updates in HIVE-12078

Whether the LLAP I/O layer is enabled. Remove property or set to false to disable LLAP I/O.

hive.llap.io.cache.orc.size

- Default Value: 1Gb
- Added In: Hive 2.0.0 with ..

Maximum size for IO allocator or ORC low-level cache.

hive.llap.io.threadpool.size

- Default Value: 10
- Added In: Hive 2.0.0 with HIVE-10081

Specify the number of threads to use for low-level IO thread pool.

hive.llap.io.orc.time.counters

- Default Value: true
- Added In: Hive 2.0.0 with HIVE-10777

Whether to enable time counters for LLAP IO layer (time spent in HDFS, etc.)

hive.llap.io.memory.mode

- Default Value: cache
- Possible Values: cache, allocator, none
- Added In: Hive 2.0.0 with HIVE-12597

LLAP IO memory usage;

'cache' (the default) uses data and metadata cache with a custom off-heap allocator,
'allocator' uses the custom allocator without the caches,
'none' doesn't use either (this mode may result in significant performance degradation)

hive.llap.io.allocator.alloc.min

- Default Value: 128Kb
- Added In: Hive 2.0.0 with HIVE-12597

Minimum allocation possible from LLAP buddy allocator. Allocations below that are padded to minimum allocation.
For ORC, should generally be the same as the expected compression buffer size, or next lowest power of 2. Must be a power of 2.

hive.llap.io.allocator.alloc.max

- Default Value: 16Mb
- Added In: Hive 2.0.0 with HIVE-12597

Maximum allocation possible from LLAP buddy allocator. For ORC, should be as large as the largest expected ORC compression buffer size. Must be a power of 2.

hive.llap.io.allocator.arena.count

- Default Value: 8
- Added In: Hive 2.0.0 with HIVE-12597

Arena count for LLAP low-level cache; cache will be allocated in the steps of (size/arena_count) bytes. This size must be <= 1Gb and >= max allocation; if it is not the case, an adjusted size will be used. Using powers of 2 is recommended.

hive.llap.io.memory.size

- Default Value: 1Gb

- Added In: Hive 2.0.0 with HIVE-12597

Maximum size for IO allocator or ORC low-level cache.

hive.llap.io.allocator.direct

- Default Value: true
- Added In: Hive 2.0.0 with HIVE-12597

Whether ORC low-level cache should use direct allocation.

hive.llap.io.allocator.nmap

- Default value: false
- Added In: Hive 2.1.0 with HIVE-13029

Whether ORC low-level cache should use memory mapped allocation (direct I/O)

hive.llap.io.allocator.nmap.path

- Default value: /tmp
- Added In: Hive 2.1.0 with HIVE-13029

The directory location for mapping NVDIMM/NVMe flash storage into the ORC low-level cache.

LLAP CBO

hive.llap.auto.allow.uber

- Default Value: true
- Added In: Hive 2.0.0 with HIVE-9777

Whether or not to allow the planner to run vertices in the AM.

hive.llap.auto.enforce.tree

- Default Value: true
- Added In: Hive 2.0.0 with HIVE-9635

Enforce that all parents are in llap, before considering vertex

hive.llap.auto.enforce.vectorized

- Default Value: true
- Added In: Hive 2.0.0 with HIVE-9635

Enforce that inputs are vectorized, before considering vertex

hive.llap.auto.enforce.stats

- Default Value: true
- Added In: Hive 2.0.0 with HIVE-9635

Enforce that column stats are available, before considering vertex.

hive.llap.auto.max.input.size

- Default Value: 10*1024*1024*1024L
- Added In: Hive 2.0.0 with HIVE-9635

Check input size, before considering vertex (-1 disables check)

hive.llap.auto.max.output.size

- Default Value: 1*1024*1024*1024L
- Added In: Hive 2.0.0 with HIVE-9635

Check output size, before considering vertex (-1 disables check)

LLAP Metrics

hive.llap.queue.metrics.percentiles.intervals

- Default Value: blank
- Added In: Hive 2.0.0 with ..

Comma-delimited set of integers denoting the desired rollover intervals (in seconds) for percentile latency metrics on the LLAP daemon producer-consumer queue.

By default, percentile latency metrics are disabled.

hive.llap.management.rpc.port

- Default Value: `15004`
- Added In: 2.0.0 with HIVE-12341

RPC port for LLAP daemon management service.

LLAP UDF Security

Whitelist based UDF support (HIVE-12852).

hive.llap.allow.permanent.fns

- Default Value: `true`
- Added In: 2.1.0 with HIVE-13307

Whether LLAP decider should allow permanent UDFs.

hive.llap.daemon.download.permanent.fns

- Default Value: `false`

- Added In: Hive 2.1.0 with HIVE-12853, HIVE-13054, and HIVE-13307

Whether LLAP daemon should localize the resources for permanent UDFs.

LLAP Security

hive.llap.daemon.keytab.file

- Default Value: (empty)
- Added In: Hive 2.0.0 with HIVE-12341

The path to the Kerberos Keytab file containing the LLAP daemon's service principal.

hive.llap.zk.sm.principal

- Default Value: (empty)
- Added In: Hive 2.0.0 with HIVE-12341

The name of the principal to use to talk to ZooKeeper for ZooKeeper SecretManager.

hive.llap.zk.sm.keytab.file

- Default Value: (empty)
- Added In: Hive 2.0.0 with HIVE-12341

The path to the Kerberos Keytab file containing the principal to use to talk to ZooKeeper for ZooKeeper SecretManager.

hive.llap.zk.sm.connectionString

- Default Value: (empty)
- Added In: Hive 2.0.0 with HIVE-12341

ZooKeeper connection string for ZooKeeper SecretManager.

hive.llap.daemon.acl

- Default Value: *
- Added In: Hive 2.0.0 with HIVE-12341 and HIVE-12813

The ACL for LLAP daemon.

hive.llap.management.acl

- Default Value: *
- Added In: Hive 2.0.0 with HIVE-12341 and HIVE-12813

The ACL for LLAP daemon management.

hive.llap.daemon.delegation.token.lifetime

- Default Value: `14d`
- Added In: Hive 2.0.0 with HIVE-12341

LLAP delegation token lifetime, in seconds if specified without a unit.

Transactions and Compactor

Hive transactions with row-level ACID functionality were added in Hive 0.13.0 (HIVE-5317 and its subtasks). For details see ACID and Transactions in Hive.

To turn on Hive transactions, change the values of these parameters from their defaults, as described below:

- **hive.txn.manager**
- **hive.compactor.initiator.on**
- **hive.compactor.worker.threads**

These parameters must also have non-default values to turn on Hive transactions:

- **hive.support.concurrency**
- **hive.enforce.bucketing** (Hive 0.x and 1.x only)
- **hive.exec.dynamic.partition.mode**

Transactions

hive.txn.manager

- Default Value: `org.apache.hadoop.hive.ql.lockmgr.DummyTxnManager`
- Hive Transactions Value: `org.apache.hadoop.hive.ql.lockmgr.DbTxnManager`
- Added In: Hive 0.13.0 with HIVE-5843

Set this to org.apache.hadoop.hive.ql.lockmgr.DbTxnManager as part of turning on Hive transactions. The default DummyTxnManager replicates pre-Hive-0.13 behavior and provides no transactions.

Turning on Hive transactions also requires appropriate settings for **hive.compactor.initiator.on**, **hive.compactor.worker.threads**, **hive.support.concurrency**, **hive.enforce.bucketing** (Hive 0.x and 1.x only), and **hive.exec.dynamic.partition.mode**.

hive.txn.strict.locking.mode

- Default Value: `true`
- Added In: Hive 2.2.0 with HIVE-15774

In strict mode non-ACID resources use standard R/W lock semantics, e.g. INSERT will acquire exclusive lock. In non-strict mode, for non-ACID resources, INSERT will only acquire shared lock, which allows two concurrent writes to the same partition but still lets lock manager prevent DROP TABLE etc. when the table is being written to. Only apples when `hive.txn.manager=org.apache.hadoop.hive.ql.lockmgr.DbTxnManager`.

hive.txn.timeout

- Default Value: `300`
- Added In: Hive 0.13.0 with HIVE-5843

Time after which transactions are declared aborted if the client has not sent a heartbeat, in seconds.

hive.txn.heartbeat.threadpool.size

- Default Value: `5`
- Added In: Hive 1.3.0 and 2.0.0 with HIVE-12366

The number of threads to use for heartbeating. For the Hive CLI one thread is enough, but HiveServer2 needs a few threads.

hive.timedout.txn.reaper.start

- Default Value: `100s`
- Added In: Hive 1.3.0 with HIVE-11317

Time delay of first reaper (the process which aborts timed-out transactions) run after the metastore start.

hive.timedout.txn.reaper.interval

- Default Value: `180s`
- Added In: Hive 1.3.0 with HIVE-11317

Time interval describing how often the reaper (the process which aborts timed-out transactions) runs.

hive.writeset.reaper.interval

- Default Value: `60s`
- Added In: Hive 1.3.0 and 2.1.0 with HIVE-13395

Frequency of WriteSet reaper runs.

hive.txn.max.open.batch

- Default Value: `1000`
- Added In: Hive 0.13.0 with HIVE-5843

Maximum number of transactions that can be fetched in one call to open_txns().

This controls how many transactions streaming agents such as Flume or Storm open simultaneously. The streaming agent then writes that number of entries into a single file (per Flume agent or Storm bolt). Thus increasing this value decreases the number of delta files created by streaming agents. But it also increases the number of open transactions that Hive has to track at any given time, which may negatively affect read performance.

hive.max.open.txns

- Default Value: `100000`
- Added In: Hive 1.3.0 and 2.1.0 with HIVE-13249

Maximum number of open transactions. If current open transactions reach this limit, future open transaction requests will be rejected, until the number goes below the limit.

hive.count.open.txns.interval

- Default Value: `1s`
- Added In: Hive 1.3.0 and 2.1.0 with HIVE-13249

Time in seconds between checks to count open transactions.

hive.txn.retryable.sqlex.regex

- Default Value: (empty)
- Added In: Hive 1.3.0 and 2.1.0 with HIVE-12637

Comma separated list of regular expression patterns for SQL state, error code, and error message of retryable SQLExceptions, that's suitable for the Hive metastore database.

For example: Can't serialize.*,40001$,^Deadlock,.*ORA-08176.*

The string that the regex will be matched against is of the following form, where ex is a SQLException:

ex.getMessage() + " (SQLState=" + ex.getSQLState() + ", ErrorCode=" + ex.getErrorCode() + ")"

Compactor

hive.compactor.initiator.on

- Default Value: `false`
- Hive Transactions Value: `true` (for exactly one instance of the Thrift metastore service)
- Added In: Hive 0.13.0 with HIVE-5843

Whether to run the initiator and cleaner threads on this metastore instance. Set this to true on one instance of the Thrift metastore service as part of turning on Hive transactions. For a complete list of parameters required for turning on transactions, see **hive.txn.manager**.

It's critical that this is enabled on exactly one metastore service instance (not enforced yet).

hive.compactor.worker.threads

- Default Value: `0`
- Hive Transactions Value: greater than `0` on at least one instance of the Thrift metastore service
- Added In: Hive 0.13.0 with HIVE-5843

How many compactor worker threads to run on this metastore instance. Set this to a positive number on one or more instances of the Thrift metastore service as part of turning on Hive transactions. For a complete list of parameters required for turning on transactions, see **hive.txn.manager**.

Worker threads spawn MapReduce jobs to do compactions. They do not do the compactions themselves. Increasing the number of worker threads will decrease the time it takes tables or partitions to be compacted once they are determined to need compaction. It will also increase the background load on the Hadoop cluster as more MapReduce jobs will be running in the background.

hive.compactor.worker.timeout

- Default Value: `86400`
- Added In: Hive 0.13.0 with HIVE-5843

Time in seconds after which a compaction job will be declared failed and the compaction re-queued.

hive.compactor.check.interval

- Default Value: `300`

- Added In: Hive 0.13.0 with HIVE-5843

Time in seconds between checks to see if any tables or partitions need to be compacted. This should be kept high because each check for compaction requires many calls against the NameNode.

Decreasing this value will reduce the time it takes for compaction to be started for a table or partition that requires compaction. However, checking if compaction is needed requires several calls to the NameNode for each table or partition that has had a transaction done on it since the last major compaction. So decreasing this value will increase the load on the NameNode.

hive.compactor.cleaner.run.interval

- Default Value: `5000`
- Added In: Hive 0.14 with HIVE-8258

Time in milliseconds between runs of the cleaner thread. Increasing this value will lengthen the time it takes to clean up old, no longer used versions of data and lower the load on the metastore server. Decreasing this value will shorten the time it takes to clean up old, no longer used versions of the data and increase the load on the metastore server.

hive.compactor.delta.num.threshold

- Default Value: `10`
- Added In: Hive 0.13.0 with HIVE-5843

Number of delta directories in a table or partition that will trigger a minor compaction.

hive.compactor.delta.pct.threshold

- Default Value: `0.1`
- Added In: Hive 0.13.0 with HIVE-5843

Percentage (fractional) size of the delta files relative to the base that will trigger a major compaction. (1.0 = 100%, so the default 0.1 = 10%.)

hive.compactor.abortedtxn.threshold

- Default Value: `1000`
- Added In: Hive 0.13.0 with HIVE-5843

Number of aborted transactions involving a given table or partition that will trigger a major compaction.

Compaction History

hive.compactor.history.retention.succeeded

- Default Value: `3`
- Added In: Hive 1.3.0 and 2.0.0 with HIVE-12353

Number of successful compaction entries to retain in history (per partition).

hive.compactor.history.retention.failed

- Default Value: `3`
- Added In: Hive 1.3.0 and 2.0.0 with HIVE-12353

Number of failed compaction entries to retain in history (per partition).

hive.compactor.history.retention.attempted

- Default Value: `2`
- Added In: Hive 1.3.0 and 2.0.0 with HIVE-12353

Number of attempted compaction entries to retain in history (per partition).

hive.compactor.history.reaper.interval

- Default Value: `2m`
- Added In: Hive 1.3.0 and 2.0.0 with HIVE-12353

Controls how often the process to purge historical record of compactions runs.

hive.compactor.initiator.failed.compacts.threshold

- Default Value: `2`
- Added In: Hive 1.3.0 and 2.0.0 with HIVE-12353

Number of consecutive failed compactions for a given partition after which the Initiator will stop attempting to schedule compactions automatically. It is still possible to use ALTER TABLE to initiate compaction. Once a manually-initiated compaction succeeds, auto-initiated compactions will resume. Note that this must be less than **hive.compactor.history.retention.failed**.

Indexing

Indexing was added in Hive 0.7.0 with HIVE-417, and bitmap indexing was added in Hive 0.8.0 with HIVE-1803. For more information see Indexing.

hive.index.compact.file.ignore.hdfs

- Default Value: `false`
- Added In: Hive 0.7.0 with HIVE-1889

When `true` the HDFS location stored in the index file will be ignored at runtime. If the data got moved or the name of the cluster got changed, the index data should still be usable.

hive.optimize.index.filter

- Default Value: `false`
- Added In: Hive 0.8.0 with HIVE-1644

Whether to enable automatic use of indexes.

hive.optimize.index.filter.compact.minsize

- Default Value: `5368709120`
- Added In: Hive 0.8.0 with HIVE-1644

Minimum size (in bytes) of the inputs on which a compact index is automatically used.

hive.optimize.index.filter.compact.maxsize

- Default Value: `-1`
- Added In: Hive 0.8.0 with HIVE-1644

Maximum size (in bytes) of the inputs on which a compact index is automatically used. A negative number is equivalent to infinity.

hive.index.compact.query.max.size

- Default Value: `10737418240`
- Added In: Hive 0.8.0 with HIVE-2096

The maximum number of bytes that a query using the compact index can read. Negative value is equivalent to infinity.

hive.index.compact.query.max.entries

- Default Value: `10000000`
- Added In: Hive 0.8.0 with HIVE-2096

The maximum number of index entries to read during a query that uses the compact index. Negative value is equivalent to infinity.

hive.exec.concatenate.check.index

- Default Value: `true`
- Added In: Hive 0.8.0 with HIVE-2125

If this sets to true, Hive will throw error when doing ALTER TABLE tbl_name [partSpec] CONCATENATE on a table/partition that has indexes on it. The reason the user want to set this to true is because it can help user to avoid handling all index drop, recreation, rebuild work. This is very helpful for tables with thousands of partitions.

hive.optimize.index.autoupdate

- Default Value: `false`
- Added In: Hive 0.8.0 with HIVE-2354

Whether or not to enable automatic rebuilding of indexes when they go stale.
Caution: Rebuilding indexes can be a lengthy and computationally expensive operation; in many cases it may be best to rebuild indexes manually.

hive.optimize.index.groupby

- Default Value: `false`
- Added In: Hive 0.8.1 with HIVE-1694

hive.index.compact.binary.search

- Default Value: `true`
- Added In: Hive 0.8.1 with HIVE-2535

Whether or not to use a binary search to find the entries in an index table that match the filter, where possible.

Statistics

See Statistics in Hive for information about how to collect and use Hive table, partition, and column statistics.

hive.stats.dbclass

- Default Value: `jdbc:derby` (Hive 0.7 to 0.12) or `fs` (Hive 0.13 and later)
- Added In: Hive 0.7 with HIVE-1361
- New Values: `counter` and `custom` added in 0.13 with HIVE-4632 and `fs` added in 0.13 with HIVE-6500

Hive 0.7 to 0.12: The default database that stores temporary Hive statistics. Options are `jdbc:derby`, `jdbc:mysql`, and `hbase` as defined in StatsSetupConst.java.

Hive 0.13 and later: The storage that stores temporary Hive statistics. In filesystem based statistics collection ("`fs`"), each task writes statistics it has collected in a file on the filesystem, which will be aggregated after the job has finished. Supported values are `fs` (filesystem), `jdbc:<database>` (where `<database>` can be `derby`, `mysql`, etc.), `hbase`, `counter`, and `custom` as defined in StatsSetupConst.java.

hive.stats.autogather

- Default Value: `true`
- Added In: Hive 0.7 with HIVE-1361

This flag enables gathering and updating statistics automatically during Hive DML operations.

> ⚠ Statistics are not gathered for `LOAD DATA` statements.

hive.stats.column.autogather

- Default Value: `false` (Hive 2.1 and later 2.x); `true` (Hive 3.0 and later)
- Added In: Hive 2.1

Extends statistics autogathering to also collect column level statistics.

hive.stats.jdbcdriver

- Default Value: `org.apache.derby.jdbc.EmbeddedDriver`
- Added In: Hive 0.7 with HIVE-1361

The JDBC driver for the database that stores temporary Hive statistics.

hive.stats.dbconnectionstring

- Default Value: `jdbc:derby:;databaseName=TempStatsStore;create=true`
- Added In: Hive 0.7 with HIVE-1361

The default connection string for the database that stores temporary Hive statistics.

hive.stats.default.publisher

- Default Value: (empty)
- Added In: Hive 0.7 with HIVE-1923

The Java class (implementing the StatsPublisher interface) that is used by default if **hive.stats.dbclass** is not JDBC or HBase (Hive 0.12.0 and earlier), or if **hive.stats.dbclass** is a custom type (Hive 0.13.0 and later: HIVE-4632).

hive.stats.default.aggregator

- Default Value: (empty)
- Added In: Hive 0.7 with HIVE-1923

The Java class (implementing the StatsAggregator interface) that is used by default if **hive.stats.dbclass** is not JDBC or HBase (Hive 0.12.0 and earlier), or if **hive.stats.dbclass** is a custom type (Hive 0.13.0 and later: HIVE-4632).

hive.stats.jdbc.timeout

- Default Value: `30`
- Added In: Hive 0.7 with HIVE-1961

Timeout value (number of seconds) used by JDBC connection and statements.

hive.stats.atomic

- Default Value: `false`
- Added In: Hive 0.7 with HIVE-1961

If this is set to true then the metastore statistics will be updated only if all types of statistics (number of rows, number of files, number of bytes, etc.) are available. Otherwise metastore statistics are updated in a best effort fashion with whatever are available.

hive.stats.retries.max

- Default Value: `0`
- Added In: Hive 0.8 with HIVE-2127

Maximum number of retries when stats publisher/aggregator got an exception updating intermediate database. Default is no tries on failures.

hive.stats.retries.wait

- Default Value: `3000`
- Added In: Hive 0.8 with HIVE-2127

The base waiting window (in milliseconds) before the next retry. The actual wait time is calculated by baseWindow * failures + baseWindow * (failure + 1) * (random number between 0.0,1.0).

hive.stats.collect.rawdatasize

- Default Value: `true`
- Added In: Hive 0.8 with HIVE-2185

If true, the raw data size is collected when analyzing tables.

hive.client.stats.publishers

- Default Value: (empty)
- Added In: Hive 0.8.1 with HIVE-2446 (patch 2)

Comma-separated list of statistics publishers to be invoked on counters on each job. A client stats publisher is specified as the name of a Java class which implements the org.apache.hadoop.hive.ql.stats.ClientStatsPublisher interface.

hive.client.stats.counters

- Default Value: (empty)
- Added In: Hive 0.8.1 with HIVE-2446 (patch 2)

Subset of counters that should be of interest for **hive.client.stats.publishers** (when one wants to limit their publishing). Non-display names should be used.

hive.stats.reliable

- Default Value: false
- Added In: Hive 0.10.0 with HIVE-1653
- New Behavior In: Hive 0.13.0 with HIVE-3777

Whether queries will fail because statistics cannot be collected completely accurately. If this is set to true, reading/writing from/into a partition or unpartitioned table may fail because the statistics could not be computed accurately. If it is set to false, the operation will succeed.

In Hive 0.13.0 and later, if **hive.stats.reliable** is false and statistics could not be computed correctly, the operation can still succeed and update the statistics but it sets a partition property "areStatsAccurate" to false. If the application needs accurate statistics, they can then be obtained in the background.

hive.stats.ndv.error

- Default Value: `20.0`
- Added In: Hive 0.10 with HIVE-1362 (patch 10)

Standard error allowed for NDV estimates, expressed in percentage. This provides a tradeoff between accuracy and compute cost. A lower value for the error indicates higher accuracy and a higher compute cost. (NDV means number of distinct values.)

hive.stats.collect.tablekeys

- Default Value: `false`
- Added In: Hive 0.10 with HIVE-3501

Whether join and group by keys on tables are derived and maintained in the QueryPlan. This is useful to identify how tables are accessed and to determine if they should be bucketed.

hive.stats.collect.scancols

- Default Value: `false`
- Added In: Hive 0.11 with HIVE-3940

Whether column accesses are tracked in the QueryPlan. This is useful to identify how tables are accessed and to determine if there are wasted columns that can be trimmed.

hive.stats.key.prefix.max.length

- Default Value: `200` (Hive 0.11 and 0.12) or `150` (Hive 0.13 and later)
- Added In: Hive 0.11 with HIVE-3750

Determines if, when the prefix of the key used for intermediate statistics collection exceeds a certain length, a hash of the key is used instead. If the value < 0 then hashing is never used, if the value >= 0 then hashing is used only when the key prefixes' length exceeds that value. The key prefix is defined as everything preceding the task ID in the key. For counter type statistics, it's maxed by **mapreduce.job.counters.group.name.max**, which is by default 128.

hive.stats.key.prefix.reserve.length

- Default Value: `24`
- Added In: Hive 0.13 with HIVE-6229

Reserved length for postfix of statistics key. Currently only meaningful for counter type statistics which should keep the length of the full statistics key smaller than the maximum length configured by **hive.stats.key.prefix.max.length**. For counter type statistics, it should be bigger than the length of LB spec if exists.

hive.stats.max.variable.length

- Default Value: `100`
- Added In: Hive 0.13 with HIVE-5369

To estimate the size of data flowing through operators in Hive/Tez (for reducer estimation etc.), average row size is multiplied with the total number of rows coming out of each operator. Average row size is computed from average column size of all columns in the row. In the absence of column statistics, for variable length columns (like string, bytes, etc.) this value will be used. For fixed length columns their corresponding Java equivalent sizes are used (float – 4 bytes, double – 8 bytes, etc.).

hive.analyze.stmt.collect.partlevel.stats

- Default Value: `true`
- Added In: Hive 0.14.0 with HIVE-7609

Prior to 0.14, on partitioned table, analyze statement used to collect table level statistics when no partition is specified. That behavior has changed beginning 0.14 to instead collect partition level statistics for all partitions. If old behavior of collecting aggregated table level statistics is desired, change the value of this config to false. This impacts only column statistics. Basic statistics are not impacted by this config.

hive.stats.list.num.entries

- Default Value: `10`
- Added In: Hive 0.13 with HIVE-5369

To estimate the size of data flowing through operators in Hive/Tez (for reducer estimation etc.), average row size is multiplied with the total number of rows coming out of each operator. Average row size is computed from average column size of all columns in the row. In the absence of column statistics and for variable length complex columns like list, the average number of entries/values can be specified using this configuration property.

hive.stats.map.num.entries

- Default Value: `10`
- Added In: Hive 0.13 with HIVE-5369

To estimate the size of data flowing through operators in Hive/Tez (for reducer estimation etc.), average row size is multiplied with the total number of rows coming out of each operator. Average row size is computed from average column size of all columns in the row. In the absence of column statistics and for variable length complex columns like map, the average number of entries/values can be specified using this configuration property.

hive.stats.map.parallelism

- Default Value: `1`
- Added In: Hive 0.13 with HIVE-5369
- Removed In: Hive 0.14 with HIVE-7156

The Hive/Tez optimizer estimates the data size flowing through each of the operators. For the GROUPBY operator, to accurately compute the data size map-side parallelism needs to be known. By default, this value is set to 1 since the optimizer is not aware of the number of mappers during compile-time. This Hive configuration property can be used to specify the number of mappers for data size computation of the GROUPBY operator. (This configuration property was removed in release 0.14.0.)

hive.stats.fetch.partition.stats

- Default Value: `true`
- Added In: Hive 0.13 with HIVE-6298
- Removed In: Hive 3.0.0 with HIVE-17932

Annotation of the operator tree with statistics information requires partition level basic statistics like number of rows, data size and file size. Partition statistics are fetched from the metastore. Fetching partition statistics for each needed partition can be expensive when the number of partitions is high. This flag can be used to disable fetching of partition statistics from the metastore. When this flag is disabled, Hive will make calls to the filesystem to get file sizes and will estimate the number of rows from the row schema.

hive.stats.fetch.column.stats

- Default Value: `false`
- Added In: Hive 0.13 with HIVE-5898

Annotation of the operator tree with statistics information requires column statistics. Column statistics are fetched from the metastore. Fetching column statistics for each needed column can be expensive when the number of columns is high. This flag can be used to disable fetching of column statistics from the metastore.

hive.stats.join.factor

- Default Value: `(float) 1.1`
- Added In: Hive 0.13 with HIVE-5921

The Hive/Tez optimizer estimates the data size flowing through each of the operators. The JOIN operator uses column statistics to estimate the number of rows flowing out of it and hence the data size. In the absence of column statistics, this factor determines the amount of rows flowing out of the JOIN operator.

hive.stats.deserialization.factor

- Default Value:
  - Hive 0.13 to 2.x.x: `(float) 1.0`
  - Hive 3.0.0 and later: `(float) 10.0`
- Added In: Hive 0.13 with HIVE-5921
- Default value changed from 1.0 to 10.0 in Hive 3.0

The Hive/Tez optimizer estimates the data size flowing through each of the operators. In the absence of basic statistics like number of rows and data size, file size is used to estimate the number of rows and data size. Since files in tables/partitions are serialized (and optionally compressed) the estimates of number of rows and data size cannot be reliably determined. This factor is multiplied with the file size to account for serialization and compression.

hive.stats.avg.row.size

- Default Value: `10000`
- Added In: Hive 0.13 with HIVE-5921

In the absence of table/partition statistics, average row size will be used to estimate the number of rows/data size.

hive.compute.query.using.stats

- Default Value: `false`
- Added In: Hive 0.13.0 with HIVE-5483

When set to true Hive will answer a few queries like min, max, and count(1) purely using statistics stored in the metastore. For basic statistics collection, set the configuration property **hive.stats.autogather** to true. For more advanced statistics collection, run ANALYZE TABLE queries.

hive.stats.gather.num.threads

- Default Value: `10`
- Added In: Hive 0.13.0 with HIVE-6578

Number of threads used by partialscan/noscan analyze command for partitioned tables. This is applicable only for file formats that implement the StatsProvidingRecordReader interface (like ORC).

hive.stats.fetch.bitvector

- Default Value: `false`
- Added In: Hive 3.0.0 with HIVE-16997

Whether Hive fetches bitvector when computing number of distinct values (ndv). Keep it set to false if you want to use the old schema without bitvectors.

Runtime Filtering

hive.tez.dynamic.semijoin.reduction

- Default Value: true
- Added In: Hive 2.2 with HIVE-15269

hive.tez.min.bloom.filter.entries

- Default Value: 1000000
- Added In: Hive 2.3 with HIVE-16260

hive.tez.max.bloom.filter.entries

- Default Value: 100000000
- Added In: Hive 2.2 with HIVE-15269

hive.tez.bloom.filter.factor

- Default Value: 2.0
- Added In: Hive 2.3 with HIVE-16260

hive.tez.bigtable.minsize.semijoin.reduction

- Default Value: 1000000
- Added In: Hive 2.3 with HIVE-16260

Authentication and Authorization

- Restricted/Hidden/Internal List and Whitelist
- Hive Client Security
- Hive Metastore Security
- SQL Standard Based Authorization

For an overview of authorization modes, see Hive Authorization.

Restricted/Hidden/Internal List and Whitelist

hive.conf.restricted.list

- Default Value:
  - Hive 0.11.0: (empty, but includes this list implicitly)
  - Hive 0.13.0: `hive.security.authenticator.manager, hive.security.authorization.manager` (HIVE-5953)
  - Hive 0.14.0: `hive.security.authenticator.manager, hive.security.authorization.manager, hive.users.in.admin.role` (HIVE-6437)
  - Hive 2.1.0: `hive.security.authenticator.manager, hive.security.authorization.manager, hive.users.in.admin.role, hive.server2.xsrf.filter.enabled` (HIVE-13853)
  - Hive 2.2.0: `hive.security.authenticator.manager, hive.security.authorization.manager, hive.security.metastore.authorization.manager, hive.security.metastore.authenticator.manager, hive.users.in.admin.role, hive.server2.xsrf.filter.enabled, hive.security.authorization.enabled` (HIVE-14099), `hive.server2.authentication.ldap.baseDN` (HIVE-15713), `hive.server2.authentication.ldap.url` (HIVE-15713), `hive.server2.authentication.ldap.Domain` (HIVE-15713), `hive.server2.authentication.ldap.groupDNPattern` (HIVE-15713), `hive.server2.authentication.ldap.groupFilter` (HIVE-15713), `hive.server2.authentication.ldap.userDNPattern` (HIVE-15713), `hive.server2.authentication.ldap.userFilter` (HIVE-15713), `hive.server2.authentication.ldap.groupMembershipKey` (HIVE-15713), `hive.server2.authentication.ldap.userMembershipKey` (HIVE-15713), `hive.server2.authentication.ldap.groupClassKey` (HIVE-15713), `hive.server2.authentication.ldap.customLDAPQuery` (HIVE-15713)
  - Hive 3.0.0: *all of the above, plus these:* `hive.spark.client.connect.timeout` (HIVE-16876), `hive.spark.client.server.connect.timeout` (HIVE-16876), `hive.spark.client.channel.log.level` (HIVE-16876), `hive.spark.client.rpc.max.size` (HIVE-16876), `hive.spark.client.rpc.threads` (HIVE-16876), `hive.spark.client.secret.bits` (HIVE-16876), `hive.spark.client.rpc.server.address` (HIVE-16876), `hive.spark.client.rpc.server.port` (HIVE-16876), `hikari.*` (HIVE-17318), `dbcp.*` (HIVE-17319)
- Added In: Hive 0.11.0 with HIVE-2935

Comma separated list of configuration properties which are immutable at runtime. For example, if **hive.security.authorization.enabled** is set to true, it should be included in this list to prevent a client from changing it to false at runtime.

hive.conf.hidden.list

- Default Value:
  - `Hive 1.2.2: javax.jdo.option.ConnectionPassword,hive.server2.keystore.password` (HIVE-9013)
  - `Hive 2.3.0: fs.s3.awsAccessKeyId,fs.s3.awsSecretAccessKey,fs.s3n.awsAccessKeyId,fs.s3n.awsSecretAccessKey,fs.s3a.access.key,fs.s3a.secre`
  - `Hive 3.0.0: dfs.adls.oauth2.credential,fs.adl.oauth2.credential` (HIVE-18228)

Comma separated list of configuration options which should not be read by normal user, such as passwords.

hive.conf.internal.variable.list

- Default Value: `hive.added.files.path,hive.added.jars.path,hive.added.archives.path`
- Added In: Hive 1.3.0 with HIVE-12346

Comma separated list of configuration options which are internally used and should not be settable via set command.

hive.security.command.whitelist

- Default Value: `set,reset,dfs,add,delete,compile[,list,reload]`
- Added In: Hive 0.13.0 with HIVE-5400 and HIVE-5252
- Changed in Hive 0.14.0 to include "list" and "reload" with HIVE-7592 ("list") and HIVE-7553 ("reload")

Comma separated list of non-SQL Hive commands that users are authorized to execute. This can be used to restrict the set of authorized commands. The supported command list is "set,reset,dfs,add,delete,compile" in Hive 0.13.0 or "set,reset,dfs,add,list,delete,reload,compile" starting in Hive 0.14.0 and by default all these commands are authorized. To restrict any of these commands, set **hive.security.command.whitelist** to a value that does not have the command in it.

Whitelist for SQL Standard Based Hive Authorization

See **hive.security.authorization.sqlstd.confwhitelist** below for information about the whitelist property that authorizes set commands in SQL standard based authorization.

Hive Client Security

hive.security.authorization.enabled

- Default Value: `false`
- Added In: Hive 0.7.0

Enable or disable the Hive client authorization.

hive.security.authorization.manager

- Default Value: `org.apache.hadoop.hive.ql.security.authorization.DefaultHiveAuthorizationProvider`
- Added In: Hive 0.7.0

The Hive client authorization manager class name. The user defined authorization class should implement interface org.apache.hadoop.hive.ql.security.authorization.HiveAuthorizationProvider.

hive.security.authenticator.manager

- Default Value: `org.apache.hadoop.hive.ql.security.HadoopDefaultAuthenticator`
- Added In: Hive 0.7.0

Hive client authenticator manager class name. The user-defined authenticator should implement interface org.apache.hadoop.hive.ql.security.HiveAuthenticationProvider.

hive.security.authorization.createtable.user.grants

- Default Value: (empty)
- Added In: Hive 0.7.0

The privileges automatically granted to some users whenever a table gets created. An example like "userX,userY:select;userZ:create" will grant select privilege to userX and userY, and grant create privilege to userZ whenever a new table created.

hive.security.authorization.createtable.group.grants

- Default Value: (empty)
- Added In: Hive 0.7.0

The privileges automatically granted to some groups whenever a table gets created. An example like "groupX,groupY:select;groupZ:create" will grant select privilege to groupX and groupY, and grant create privilege to groupZ whenever a new table created.

hive.security.authorization.createtable.role.grants

- Default Value: (empty)
- Added In: Hive 0.7.0

The privileges automatically granted to some roles whenever a table gets created. An example like "roleX,roleY:select;roleZ:create" will grant select privilege to roleX and roleY, and grant create privilege to roleZ whenever a new table created.

hive.security.authorization.createtable.owner.grants

- Default Value: (empty)
- Added In: Hive 0.7.0

The privileges automatically granted to the owner whenever a table gets created. An example like "select,drop" will grant select and drop privilege to the owner of the table. Note that the default gives the creator of a table no access to the table.

Hive Metastore Security

Metastore-side security was added in Hive 0.10.0 (HIVE-3705). For more information, see the overview in Authorization and details in Storage Based Authorization in the Metastore Server.

For general metastore configuration properties, see MetaStore.

hive.metastore.pre.event.listeners

- Default Value: (empty)
- Added In: Hive 0.9.0 with HIVE-2853

The pre-event listener classes to be loaded on the metastore side to run code whenever databases, tables, and partitions are created, altered, or dropped. Set this configuration property to `org.apache.hadoop.hive.ql.security.authorization.AuthorizationPreEventListener` in hive-site.xml to turn on Hive metastore-side security.

hive.security.metastore.authorization.manager

- Default Value: `org.apache.hadoop.hive.ql.security.authorization.DefaultHiveMetastoreAuthorizationProvider`
- Added In: Hive 0.10.0 with HIVE-3705; revised in Hive 0.14.0 with HIVE-7209

Hive 0.13 and earlier: The authorization manager class name to be used in the metastore for authorization. The user-defined authorization class should implement interface `org.apache.hadoop.hive.ql.security.authorization.HiveMetastoreAuthorizationProvider`.

Hive 0.14 and later: Names of authorization manager classes (comma separated) to be used in the metastore for authorization. User-defined authorization classes should implement interface org.apache.hadoop.hive.ql.security.authorization.HiveMetastoreAuthorizationProvider. All authorization manager classes have to successfully authorize the metastore API call for the command execution to be allowed.

The DefaultHiveMetastoreAuthorizationProvider implements the standard Hive grant/revoke model. A storage-based authorization implementation is also provided to use as the value of this configuration property:

- `org.apache.hadoop.hive.ql.security.authorization.StorageBasedAuthorizationProvider`

which uses HDFS permissions to provide authorization instead of using Hive-style grant-based authorization.

hive.security.metastore.authenticator.manager

- Default Value: `org.apache.hadoop.hive.ql.security.HadoopDefaultMetastoreAuthenticator`
- Added In: Hive 0.10.0 with HIVE-3705

The authenticator manager class name to be used in the metastore for authentication. The user-defined authenticator class should implement interface `org.apache.hadoop.hive.ql.security.HiveAuthenticationProvider`.

hive.security.metastore.authorization.auth.reads

- Default Value: `true`
- Added In: Hive 0.14.0 with HIVE-8221

If this is true, the metastore authorizer authorizes read actions on database and table. See Storage Based Authorization.

hive.metastore.token.signature

- Default Value: "" (empty string)
- Added in Hive 0.7.0, added to HiveConf in Hive 2.1.0

The delegation token service name to match when selecting a token from the current user's tokens.

SQL Standard Based Authorization

> **Version**
> Hive 0.13.0 introduces fine-grained authorization based on the SQL standard authorization model. See HIVE-5837 for the functional specification and list of subtasks.

hive.users.in.admin.role

- Default Value: (empty)
- Added In: Hive 0.13.0 with HIVE-5959

A comma separated list of users which will be added to the ADMIN role when the metastore starts up. More users can still be added later on.

hive.security.authorization.sqlstd.confwhitelist

- Default Value: (empty, but includes list shown below implicitly)
- Added In: Hive 0.13.0 with HIVE-6846; updated in Hive 0.14.0 with HIVE-8534 and in subsequent releases with several JIRA issues

A Java regex. Configuration properties that match this regex can be modified by user when SQL standard base authorization is used.

If this parameter is not set, the default list is added by the SQL standard authorizer. To display the default list for the current release, use the command '`set hive.security.authorization.sqlstd.confwhitelist`'.

In Hive 0.13.0 the default white list has these properties (see HIVE-6846 for the same list arranged one property per line):

> hive.exec.reducers.bytes.per.reducer, hive.exec.reducers.max, hive.map.aggr, hive.map.aggr.hash.percentmemory, hive.map.aggr.hash.force.flush.memory.threshold, hive.map.aggr.hash.min.reduction, hive.groupby.skewindata, hive.optimize.multigroupby.common.distincts, hive.optimize.index.groupby, hive.optimize.ppd, hive.optimize.ppd.storage, hive.ppd.recognizetransivity, hive.optimize.groupby, hive.optimize.sort.dynamic.partition, hive.optimize.union.remove, hive.multigroupby.singlereducer, hive.map.groupby.sorted, hive.map.groupby.sorted.testmode, hive.optimize.skewjoin, hive.optimize.skewjoin.compiletime, hive.mapred.mode, hive.enforce.bucketmapjoin, hive.exec.compress.output, hive.exec.compress.intermediate, hive.exec.parallel, hive.exec.parallel.thread.number, hive.exec.rowoffset, hive.merge.mapfiles, hive.merge.mapredfiles, hive.merge.tezfiles, hive.ignore.mapjoin.hint, hive.auto.convert.join, hive.auto.convert.join.noconditionaltask, hive.auto.convert.join.noconditionaltask.size, hive.auto.convert.join.use.nonstaged, hive.enforce.bucketing, hive.enforce.sorting, hive.enforce.sortmergebucketmapjoin, hive.auto.convert.sortmerge.join, hive.execution.engine, hive.vectorized.execution.enabled, hive.mapjoin.optimized.keys, hive.mapjoin.lazy.hashtable, hive.exec.check.crossproducts, hive.compat, hive.exec.dynamic.partition.mode, mapred.reduce.tasks, mapred.output.compression.codec, mapred.map.output.compression.codec, mapreduce.job.reduce.slowstart.completedmaps, mapreduce.job.queuename.

> **Version Information**
> Hive 0.14.0 adds new parameters to the default white list (see HIVE-8534).
>
> Hive 1.1.0 removes some parameters (see HIVE-9331).
>
> Hive 1.2.0 and 1.2.1 add more new parameters (see HIVE-10578, HIVE-10678, and HIVE-10967).
>
> Hive 1.3.0, 2.1.1, and 2.2.0 add further new parameters (see HIVE-14073).
>
> Hive 3.0.0 fixes a parameter added in 1.2.1, changing mapred.job.queuename to mapred.job.queue.name (see HIVE-17584).

Some parameters are added automatically when they match one of the regex specifications for the white list in HiveConf.java (for example, **hive.log.trace.id** in Hive 2.0.0 – see HIVE-12419).

Note that the **hive.conf.restricted.list** checks are still enforced after the white list check.

hive.security.authorization.sqlstd.confwhitelist.append

- Default Value: (empty)
- Added In: Hive 0.14.0 with HIVE-8534

Second Java regex that the whitelist of configuration properties would match in addition to **hive.security.authorization.sqlstd.confwhitelist**. Do not include a starting | in the value.

Using this regex instead of updating the original regex for **hive.security.authorization.sqlstd.confwhitelist** means that you can append to the default that is set by SQL standard authorization instead of replacing it entirely.

hive.server2.builtin.udf.whitelist

- Default Value: (empty, treated as not set – all UDFs allowed)
- Added In: Hive 1.1.0 with HIVE-8893

A comma separated list of builtin UDFs that are allowed to be executed. A UDF that is not included in the list will return an error if invoked from a query. If set to empty, then treated as wildcard character – all UDFs will be allowed. Note that this configuration is read at the startup time by HiveServer2 and changing this using a 'set' command in a session won't change the behavior.

hive.server2.builtin.udf.blacklist

- Default Value: (empty)
- Added In: Hive 1.1.0 with HIVE-8893

A comma separated list of builtin UDFs that are not allowed to be executed. A UDF that is included in the list will return an error if invoked from a query. Note that this configuration is read at the startup time by HiveServer2 and changing this using a 'set' command in a session won't change the behavior.

hive.security.authorization.task.factory

- Default Value: `org.apache.hadoop.hive.ql.parse.authorization.HiveAuthorizationTaskFactoryImpl`
- Added In: Hive 1.1.0 with HIVE-8611

To override the default authorization DDL handling, set **hive.security.authorization.task.factory** to a class that implements the org.apache.hadoop.hive.ql.parse.authorization.HiveAuthorizationTaskFactory interface.

Archiving

See Archiving for File Count Reduction for general information about Hive support for Hadoop archives.

fs.har.impl

- Default Value: `org.apache.hadoop.hive.shims.HiveHarFileSystem`
- Added In: Hive 0.8.1

The implementation for accessing Hadoop Archives. Note that this won't be applicable to Hadoop versions less than 0.20.

hive.archive.enabled

- Default Value: `false`
- Added In: Hive 0.6.0

Whether archiving operations are permitted.

hive.archive.har.parentdir.settable

- Default Value: `false`
- Added In: Hive 0.6.0
- Removed In: Hive 0.10.0 with HIVE-3338

In new Hadoop versions, the parent directory must be set while creating a HAR. Because this functionality is hard to detect with just version numbers, this configuration variable needed to be set manually in Hive releases 0.6.0 through 0.9.0. (This configuration property was removed in release 0.10.0.)

Locking

See Hive Concurrency Model for general information about locking.

hive.support.concurrency

- Default Value: `false`
- Added In: Hive 0.7.0 with HIVE-1293

Whether Hive supports concurrency or not. A ZooKeeper instance must be up and running for the default Hive lock manager to support read-write locks.

Set to `true` to support INSERT ... VALUES, UPDATE, and DELETE transactions (Hive 0.14.0 and later). For a complete list of parameters required for turning on Hive transactions, see **hive.txn.manager**.

hive.lock.manager

- Default Value: `org.apache.hadoop.hive.ql.lockmgr.zookeeper.ZooKeeperHiveLockManager`
- Added In: Hive 0.7.0 with HIVE-1293

The lock manager to use when **hive.support.concurrency** is set to `true`.

hive.lock.mapred.only.operation

- Default Value: `false`
- Added In: Hive 0.8.0

This configuration property is to control whether or not only do lock on queries that need to execute at least one mapred job.

hive.lock.query.string.max.length

- Default Value: 1000000
- Added In: Hive 3.0.0

The maximum length of the query string to store in the lock. The default value is 1000000, since the data limit of a znode is 1MB

hive.lock.numretries

- Default Value: `100`
- Added In: Hive 0.7.0 with HIVE-1293

The total number of times you want to try to get all the locks.

hive.unlock.numretries

- Default Value: `10`
- Added In: Hive 0.8.1

The total number of times you want to do one unlock.

hive.lock.sleep.between.retries

- Default Value: `60`
- Added In: Hive 0.7.0 with HIVE-1293

The sleep time (in seconds) between various retries.

hive.zookeeper.quorum

- Default Value: (empty)
- Added In: Hive 0.7.0 with HIVE-1293

The list of ZooKeeper servers to talk to. This is only needed for read/write locks.

hive.zookeeper.client.port

- Default Value:
  - Hive 0.7.0: (empty)
  - Hive 0.8.0 and later: `2181` (HIVE-2196)
- Added In: Hive 0.7.0 with HIVE-1293

The port of ZooKeeper servers to talk to. This is only needed for read/write locks.

hive.zookeeper.session.timeout

- Default Value:
  - Hive 0.7.0 to 1.1.x: `600000ms`
  - Hive 1.2.0 and later: `1200000ms` (HIVE-8890)
- Added In: Hive 0.7.0 with HIVE-1293

ZooKeeper client's session timeout (in milliseconds). The client is disconnected, and as a result, all locks released, if a heartbeat is not sent in the timeout.

hive.zookeeper.namespace

- Default Value: `hive_zookeeper_namespace`
- Added In: Hive 0.7.0

The parent node under which all ZooKeeper nodes are created.

hive.zookeeper.clean.extra.nodes

- Default Value: `false`
- Added In: Hive 0.7.0

Clean extra nodes at the end of the session.

hive.lockmgr.zookeeper.default.partition.name

- Default Value: `__HIVE_DEFAULT_ZOOKEEPER_PARTITION__`
- Added In: Hive 0.7.0 with HIVE-1293

The default partition name when ZooKeeperHiveLockManager is the **hive lock manager**.

Metrics

The metrics that Hive collects can be viewed in the HiveServer2 Web UI. For more information, see Hive Metrics.

hive.metastore.metrics.enabled

- Default Value: `false`
- Added in: Hive 1.3.0 and 2.0.0 with HIVE-10761

Enable metrics on the Hive Metastore Service. (For other metastore configuration properties, see the Metastore and Hive Metastore Security sections.)

hive.server2.metrics.enabled

- Default Value: `false`
- Added in: Hive 1.3.0 and 2.0.0 with HIVE-10761

Enable metrics on HiveServer2. (For other HiveServer2 configuration properties, see the HiveServer2 section.)

hive.service.metrics.class

- Default Value: `org.apache.hadoop.hive.common.metrics.metrics2.CodahaleMetrics`
- Added in: Hive 1.3.0 and 2.0.0 with HIVE-10761

Hive metrics subsystem implementation class. "org.apache.hadoop.hive.common.metrics.metrics2.CodahaleMetrics" is the new implementation. To revert back to the old implementation before Hive 1.3 and 2.0 along with its built-in JMX reporting capabilities, choose "org.apache.hadoop.hive.common.metrics.LegacyMetrics".

hive.service.metrics.reporter

- Default Value: `"JSON_FILE, JMX"`
- Added in: Hive 1.3.0 and 2.0.0 with HIVE-10761
- Deprecated in: Hive 3.0.0 with HIVE-16206

Reporter type for metric class org.apache.hadoop.hive.common.metrics.metrics2.CodahaleMetrics, a comma separated list of values JMX, CONSOLE, JSON_FILE.

A new reporter type HADOOP2 has been added in Hive 2.1.0 with HIVE-13480.

hive.service.metrics.codahale.reporter.classes

- Default Value: "org.apache.hadoop.hive.common.metrics.metrics2.JsonFileMetricsReporter, org.apache.hadoop.hive.common.metrics.metrics2.JmxMetricsReporter"
- Added in: Hive 3.0.0 with HIVE-16206

Comma separated list of reporter implementation classes for metric class org.apache.hadoop.hive.common.metrics.metrics2.CodahaleMetrics. Overrides hive.service.metrics.reporter conf if present.

hive.service.metrics.file.location

- Default Value: `"/tmp/report.json"`
- Added in: Hive 1.3.0 and 2.0.0 with HIVE-10761

For **hive.service.metrics.class** org.apache.hadoop.hive.common.metrics.metrics2.CodahaleMetrics and **hive.service.metrics.reporter** JSON_FILE, this is the location of the local JSON metrics file dump. This file will get overwritten at every interval of **hive.service.metrics.file.frequency**.

hive.service.metrics.file.frequency

- Default Value: 5 seconds
- Added in: Hive 1.3.0 and 2.0.0 with HIVE-10761

For **hive.service.metrics.class** org.apache.hadoop.hive.common.metrics.metrics2.CodahaleMetrics and **hive.service.metrics.reporter** JSON_FILE, this is the frequency of updating the JSON metrics file.

hive.service.metrics.hadoop2.component

- Default Value: `"hive"`
- Added in: Hive 2.1.0 with HIVE-13480

For **hive.service.metrics.class** org.apache.hadoop.hive.common.metrics.metrics2.CodahaleMetrics and **hive.service.metrics.reporter** HADOOP2, this is the component name to provide to the HADOOP2 metrics system. Ideally 'hivemetastore' for the MetaStore and 'hiveserver2' for HiveServer2. The metrics will be updated at every interval of **hive.service.metrics.hadoop2.frequency**.

hive.service.metrics.hadoop2.frequency

- Default Value: 30 seconds
- Added in: Hive 2.1.0 with HIVE-13480

For **hive.service.metrics.class** org.apache.hadoop.hive.common.metrics.metrics2.CodahaleMetrics and **hive.service.metrics.reporter** HADOOP2, this is the frequency of updating the HADOOP2 metrics system.

Clustering

hive.cluster.delegation.token.store.class

- Default Value: `org.apache.hadoop.hive.thrift.MemoryTokenStore`
- Added In: Hive 0.9.0

The delegation token store implementation. Set to org.apache.hadoop.hive.thrift.ZooKeeperTokenStore for load-balanced cluster.

hive.cluster.delegation.token.store.zookeeper.connectString

- Default Value: `localhost:2181`
- Added In: Hive 0.9.0

The ZooKeeper token store connect string.

hive.cluster.delegation.token.store.zookeeper.znode

- Default Value: `/hive/cluster/delegation`
- Added In: Hive 0.9.0

The root path for token store data.

hive.cluster.delegation.token.store.zookeeper.acl

- Default Value: `sasl:hive/host1@EXAMPLE.COM:cdrwa,sasl:hive/host2@EXAMPLE.COM:cdrwa`
- Added In: Hive 0.9.0

ACL for token store entries. List comma separated all server principals for the cluster.

Regions

> **Reverted by HIVE-2612 in Hive 0.9.0**
> The configuration properties that used to be documented in this section (hive.use.input.primary.region, hive.default.region.name, and hive.region.properties) existed temporarily in trunk before Hive release 0.9.0 but they were removed before the release. See HIVE-2612 and HIVE-2965.
>
> Apologies for any confusion caused by their former inclusion in this document.

Command Line Interface

hive.cli.print.header

- Default Value: `false`
- Added In: Hive 0.7.0

Whether to print the names of the columns in query output.

hive.cli.print.current.db

- Default Value: `false`
- Added In: Hive 0.8.1

Whether to include the current database in the Hive prompt.

HBase StorageHandler

hive.hbase.wal.enabled

- Default Value: `true`
- Added In: Hive 0.6.0 with HIVE-1383

Whether writes to HBase should be forced to the write-ahead log. Disabling this improves HBase write performance at the risk of lost writes in case of a crash.

hive.hbase.generatehfiles

- Default Value: false
- Added In: Hive 0.14.0 with HIVE-6473 and HIVE-7211

True when HBaseStorageHandler should generate hfiles instead of operate against the online table.

Hive Web Interface (HWI) (component removed as of Hive 2.2.0)

hive.hwi.war.file

- Default Value: `lib/hive-hwi-<version>.war`
- Added In: Hive 0.3.0 with default `lib/hive_hwi.war`, default changed to `lib/hive-hwi-<version>.war` in Hive 0.5 (HIVE-978 and HIVE-1183)
- Removed In: Hive 2.2.0 with HIVE-15622

This sets the path to the HWI war file, relative to `${HIVE_HOME}`. (This configuration property was removed in release 2.2.0.)

hive.hwi.listen.host

- Default Value: `0.0.0.0`
- Added In: Hive 0.3.0
- Removed In: Hive 2.2.0 with HIVE-15622

This is the host address the Hive Web Interface will listen on. (This configuration property was removed in release 2.2.0.)

hive.hwi.listen.port

- Default Value: `9999`
- Added In: Hive 0.3.0
- Removed In: Hive 2.2.0 with HIVE-15622

This is the port the Hive Web Interface will listen on. (This configuration property was removed in release 2.2.0.)

Replication

hive.repl.rootdir

- Default Value: `/usr/hive/repl/`
- Added in: Hive 2.2.0 with HIVE-15151

This is an HDFS root directory under which Hive's REPL DUMP command will operate, creating dumps to replicate along to other warehouses.

hive.repl.replica.functions.root.dir

- Default Value: `/usr/hive/repl/functions`
- Added in: Hive 3.0.0 with HIVE-16591

Root directory on the replica warehouse where the repl sub-system will store jars from the primary warehouse.

hive.repl.partitions.dump.parallelism

- Default Value: `100`
- Added in: Hive 3.0.0 with HIVE-16895; default changed with HIVE-17625 (also in 3.0.0)

Number of threads that will be used to dump partition data information during REPL DUMP.

hive.repl.approx.max.load.tasks

- Default Value: `10000`
- Added in: Hive 3.0.0 with HIVE-16896

Provide an approximation of the maximum number of tasks that should be executed before dynamically generating the next set of tasks. The number is approximate as Hive will stop at a slightly higher number, the reason being some events might lead to a task increment that would cross the specified limit.

hive.repl.dump.metadata.only

- Default Value: `false`
- Added in: Hive 3.0.0 with HIVE-18352

Indicates whether the REPL DUMP command dumps only metadata information (`true`) or data + metadata (`false`).

hive.repl.dump.include.acid.tables

- Default Value: `false`
- Added in: Hive 3.0.0 with HIVE-18352

Indicates whether replication dump should include information about ACID tables. It should be used in conjunction with **hive.repl.dump.metadata.only** to enable copying of metadata for ACID tables which do not require the corresponding transaction semantics to be applied on target. This can be removed when ACID table replication is supported.

hive.repl.add.raw.reserved.namespace

- Default Value: `false`
- Added in: Hive 3.0.0 with HIVE-18341

For TDE with same encryption keys on source and target, allow Distcp super user to access the raw bytes from filesystem without decrypting on source and then encrypting on target.

Blobstore (i.e. Amazon S3)

Starting in release 2.2.0, a set of configurations was added to enable read/write performance improvements when working with tables stored on blobstore systems, such as Amazon S3.

hive.blobstore.supported.schemes

- Default value: `s3,s3a,s3n`
- Added In: Hive 2.2.0 with HIVE-14270

List of supported blobstore schemes that Hive uses to apply special read/write performance improvements.

hive.blobstore.optimizations.enabled

- Default value: `true`
- Added In: Hive 2.2.0 with HIVE-15121

This parameter is a global variable that enables a number of optimizations when running on blobstores.
Some of the optimizations, such as **hive.blobstore.use.blobstore.as.scratchdir**, won't be used if this variable is set to false.

hive.blobstore.use.blobstore.as.scratchdir

- Default value: `false`
- Added In: Hive 2.2.0 with HIVE-14270

Set this to true to enable the use of scratch directories directly on blob storage systems (it may cause performance penalties).

hive.exec.input.listing.max.threads

- Default value: `0` (disabled)
- Added In: Hive 2.2.0 with HIVE-15881

Set this to a maximum number of threads that Hive will use to list file information from file systems, such as file size and number of files per table (recommended > 1 for blobstore).

Test Properties

Note: This is an incomplete list of configuration properties used by developers when running Hive tests. For other test properties, search for "hive.test." in hive-default.xml.template or HiveConf.java. Also see Beeline Query Unit Test.

hive.test.mode

- Default Value: `false`
- Added In: Hive 0.4.0

Whether Hive is running in test mode. If yes, it turns on sampling and prefixes the output tablename.

hive.test.mode.prefix

- Default Value: `test_`
- Added In: Hive 0.4.0

If Hive is running in test mode, prefixes the output table by this string.

hive.test.mode.samplefreq

- Default Value: `32`
- Added In: Hive 0.4.0

If Hive is running in test mode and table is not bucketed, sampling frequency.

hive.test.mode.nosamplelist

- Default Value: (empty)
- Added In: Hive 0.4.0

If Hive is running in test mode, don't sample the above comma separated list of tables.

hive.exec.submit.local.task.via.child

- Default Value: `true`
- Added In: Hive 0.14.0 with HIVE-7271

Determines whether local tasks (typically mapjoin hashtable generation phase) run in a separate JVM (`true` recommended) or not. Avoids the overhead of spawning new JVMs, but can lead to out-of-memory issues. A `false` setting is only useful when running unit tests. See HIVE-7271 for details.

HCatalog Configuration Properties

Starting in Hive release 0.11.0, HCatalog is installed and configured with Hive. The HCatalog server is the same as the Hive metastore. See Hive Metastore Administration for metastore configuration properties. For Hive releases prior to 0.11.0, see the "Thrift Server Setup" section in the HCatalog 0.5.0 document Installation from Tarball for information about setting the Hive metastore configuration properties.

Jobs submitted to HCatalog can specify configuration properties that affect storage, error tolerance, and other kinds of behavior during the job. See HCatalog Configuration Properties for details.

WebHCat Configuration Properties

For WebHCat configuration, see Configuration Variables in the WebHCat manual.

No labels

Powered by a free **Atlassian Confluence Open Source Project License** granted to Apache Software Foundation. Evaluate Confluence today.
This Confluence installation runs a Free Gliffy License - Evaluate the Gliffy Confluence Plugin for your Wiki!