

Version 5.0 Release Notes

What's New in Version 5.0.0

The 5.0.0 release of the MapR Distribution for Apache Hadoop contains the following new features. If you are upgrading from Version 3.1.x or earlier, see also the Release Notes for Versions 4.0.1, 4.0.2, and 4.1.0.

Auditing

The auditing feature in MapR logs the following events in audit logs:

- Operations on filesystem objects (directories, files, and MapR-DB tables), as well as accesses of these objects
- Administrative operations on clusters, including the execution of maprccli commands

Log entries are written in JSON-format and can be queried or processed by Drill and other third-party tools. Log files are also retained for as long as you specify.

By analyzing audit records, security analysts can answer questions such as these:

- Who touched customer records outside of business hours?
- What actions did users take in the days before leaving the company?
- What operations were performed without following change control?
- Are users accessing sensitive files from protected or secured IP addresses?
- Why do my reports look different, despite sourcing from the same underlying data?

Data scientists can analyze audit records to find out answers to questions such as these:

- Which data is used most frequently, is therefore of high value, and should be shared more broadly?
- Which data is least commonly used, is therefore of low value, and could be purged?
- Which data should be used more, is therefore underused, and needs better advertising?
- Which administrative actions are most commonly performed and are therefore candidates for automation?

Indexing of MapR-DB Tables in Elasticsearch

You can create external indexes for your MapR-DB data by indexing columns, column families, or entire tables in Elasticsearch.

When client applications update data in a source table, MapR-DB replicates the update to the Elasticsearch type that is associated with it.

Updates to indexes happen in near real-time because individual updates to your MapR-DB source tables are replicated to Elasticsearch. There is no batching of updates, which would cause recurring situations in which data would be available in MapR-DB but not searchable in your indexes. Therefore, there is minimal latency between the availability of data in MapR-DB and the searchability of that data by end users.

By default, MapR-DB data is converted to strings when indexed. However, you can have MapR-DB convert your data to a number of supported data types, or you can code your own custom data conversions.

The MapR distribution does not include Elasticsearch, which you can get from <https://www.elastic.co/>. This indexing feature of MapR-DB works with Elasticsearch version 1.4.

YARN Enhancements

MapR 5.0 runs the 2.7.0 version of Hadoop. Hadoop 2.7.0 introduces some notable enhancements and a number of bug fixes.

MapR 5.0 includes the following enhancements:

- **Work-preserving restarts of ResourceManager**
Applications are not shut down at the point at which the ResourceManager restarts. After a ResourceManager restarts, it can resume the processing of applications that were in progress. Previously, the ResourceManager would shut down and restart applications that were running at the time that the ResourceManager required a restart.
- **Container-preserving restart of NodeManager**
Containers remain active during the restart of the NodeManager. Previously, containers that were available at that the time that NodeManager required a restart were shut down and were re-allocated after the restart of the NodeManager.
- **Pluggable YARN authorization**
A common interface was introduced for YARN authorization for queues and ACLs.
- **Ability to limit the number of concurrent tasks for a MapReduce V2 job**
You can limit the number of concurrent tasks for a MapReduce job by configuring the following properties in the `mapred-site.xml`:
 - `mapreduce.job.running.map.limit`
 - `mapreduce.job.running.reduce.limit`

By default, the properties are set to 0 (no limit)

- **Increased performance of FileOutputCommitter for very large jobs with many output files**

For more information, see [MAPREDUCE-4815](#).

- **Windows Azure Blob Storage**

You can read and write data to Azure Blob Storage. For more information, see <http://hadoop.apache.org/docs/r2.7.0/hadoop-azure/index.html>

MapR 5.0 also includes the following bug fixes, which became available after Hadoop 2.7.0 was released:

- [MAPREDUCE-6353](#): Divide by zero error in MR AM when calculating available containers
- [YARN-3351](#): AppMaster tracking URL is broken in HA
- [HADOOP-11872](#): On Windows clients, the `hadoop fs` command prints an incorrect message that says to use `yarn jar` instead of `hadoop fs`.

The following Hadoop 2.6 features are not supported by MapR:

- Service Registry for applications
- Node labels during scheduling (MapR has its own label-based scheduling feature.)
- Global, shared cache for application artifacts
- Running of applications natively in Docker containers
- Time-based resource reservations in the Capacity Scheduler

The following Hadoop 2.7 features are not supported by MapR:

- Automatic shared, global caching of YARN localized resources
- HDFS variable-length blocks
MapR clusters use MapR-FS instead of HDFS. In MapR-FS, users can set block size per directory.
- HDFS quotas per storage type
MapR supports quotas that can be set for each volume or entity (user or group). Quotas have been available with MapR since MapR 1.x.
- HDFS file truncate API

Volume Upgrades for Read-Write Mirror Support

The feature that allows mirror volumes to be promoted to read-write was introduced in Version 4.0.2; however, this feature worked only for new-format volumes: that is, volumes that were created in new 4.0.2 or 4.1 clusters. In Version 5.0, you can upgrade old volumes to the new format by using the `maprcli volume upgradeformat` command. After running this command, you can use the volume as a read-only or read-write volume by following standard procedures. New volumes that you create in Version 5.0 are created in the new format and are promotable automatically.

Automatic Volume Management

This feature allows sub-volumes (children) to inherit properties from their parent volume. The `maprcli volume create` and `volume modify` commands provide parameters for setting the inheritance feature.

Replication Factor for Name Container

This feature allows you to increase the replication factor of the name container independent of the rest of the data containers. The desired (NS Replication) and minimum (Min NS Replication) parameters are the namespace replication factors. The NS Replication factor is the desired number of replicated copies for the namespace container. The Min NS Replication factor is the smallest number of replicated copies before re-replication occurs.

Filter Support in the MapR-DB C APIs

You can use two new APIs -- `hb_get_set_filter()` and `hb_scanner_set_filter()` -- to filter the results of GET and SCAN operations in C applications that access MapR-DB tables.

Support for the HBase Java API `checkAndMutate()`

In Java applications that access MapR-DB tables, you can use the `checkAndMutate()` API in the `HTable` class to check whether the value of a row, column family, or qualifier matches an expected value.

Installation and Upgrade Notes

Before installing or upgrading to Version 5.0, read the following sections.

MapR Installer Updates

You can install MapR Version 5.0 through the web interface provided by the MapR Installer.

Rolling Upgrades

Rolling upgrades are supported from all 4.x versions to Version 5.0.

Enabling New Features

If you upgraded your MapR cluster from Version 4.1 or earlier, you must enable the auditing feature. Run this command:

```
{maprcli config save -values {"mfs.feature.audit.support":"1"}}
```

If you upgraded your MapR cluster from Version 4.1 or earlier and you want to enable full support for promotable mirrors, run the following commands:

```
maprcli config save -values {"mfs.feature.volume.upgrade":"1"}
maprcli config save -values {"mfs.feature.rwmirror.support":"1"}
```

For more information about promotable mirror support, see the complete 5.0 documentation.

To verify that new 5.0 features are enabled, run this command for each feature:

```
maprcli config load -json | grep "mfs.feature.<feature-name>"
```

If you are upgrading from Version 4.0.2 or earlier, you may need to enable additional features that were added in previous releases, such as Version 4.1.

MapR Client Compatibility

In general, Version 4.0.1, 4.0.2, and 4.1 MapR clients will continue to work against a cluster that is upgraded to Version 5.0.0. However, the RM HA configuration on the client must match the configuration on the cluster. For example, zero-configuration RM HA was not supported in Version 4.0.1 so a Version 4.0.1 YARN client will not work with a Version 5.0.0 RM HA cluster.

MapR Interoperability Matrix

See the Interoperability Matrix pages for detailed information about MapR server, JDK, client, and ecosystem compatibility.

Ecosystem Support

The Hadoop ecosystem components are hosted in a repository that is specific to Version 5.x: <http://package.mapr.com/releases/ecosystem-5.x>

Version 5.0.0 works with the following ecosystem projects:

Project	Versions Supported in MapR 5.0.0
Hive	0.13, 1.0

Spark	1.3.1
Pig	0.14
Flume	1.5
Sqoop	1.4.6
Sqoop2	1.99.3
Mahout	0.10
Oozie	4.1.0
Hue	3.7
HBase	0.98.9, 0.98.12.1
Drill	1.1
Asynchbase	1.6
Cascading	2.5
HTTPFS	1
Tez	0.5.3
MapReduce	1.0.3
Storm	0.9.4
Sentry	1.4

Known Issues

You may encounter the following known issues after upgrading to Version 5.0.0.

Services Do Not Start After CentOS 7.0 Reboot

This problem is specific to the CentOS 7.0 operating system. When a CentOS 7.0 node is rebooted, Warden starts but fails to bring up other cluster services.

To work around this problem, manually add the fully qualified domain name to the `/etc/hosts` file before rebooting the system.

Issue with Removing Replicas When Indexing MapR-DB Data in Elasticsearch

Only one user should manage indexing of any given source MapR-DB table. If indexing of the table in a given Elasticsearch type is no longer needed and any other user attempts to run the command `maprcli table replica elasticsearch remove` to stop replicating from the table to that Elasticsearch type, the command will fail with the message that permission is denied.

Metrics Database Not Yet Supported for YARN Applications

You cannot use the Metrics Database to record activity for applications that run in YARN (MRv2). The database only supports MRv1 jobs.

Resource Manager Issues

14696/15100: When automatic or manual ResourceManager failover is enabled and a job is submitted with impersonation turned ON by a user without impersonation privileges, the job submission eventually times out instead of returning an appropriate error. This behavior does not affect standard ecosystem services such as HiveServer because they are configured to run as the `mapr` user (with impersonation allowed). However, this problem does affect non-ecosystem applications or services that attempt to submit jobs with impersonation turned ON. MapR recommends that customers add the user in question to the impersonation list so that the job can proceed.

14907: When several jobs are submitted and the ResourceManager is using the `ZKRMStateStore` for failover, the cluster may experience

ZooKeeper timeouts and instability. MapR recommends that customers always use the FileSystemRMStateStore to support ResourceManager HA. See Configuring the ResourceManager State Store.

Installation and Configuration Issues

CentOS Version 6.3 and Earlier: MapR installations on Version 6.3 and earlier may fail because of an unresolved dependency on the redhat-lsb-core package.

This problem occurs when the CentOS repository points to vault.centos.org, rather than mirror.centos.org. If you encounter this problem, use one of the following workarounds:

- Add this repository: http://mirror.centos.org/centos/6/os/x86_64/
- Manually download and install the RPM:
 1. wget http://mirror.centos.org/centos/6/os/x86_64/Packages/redhat-lsb-core-4.0-7.el6.centos.x86_64.rpm
 2. yum localinstall redhat-lsb-core-4.0-7.el6.centos.x86_64.rpm

16216: When you run `configure.sh` with the `-HS` option on client nodes, the `mapred-site.xml` is re-generated and does not retain existing user settings. To work around this problem, use the `-R` option in the command.

16155: In order to reconfigure a Mac client from secure mode to non-secure mode (or vice versa), you must follow these steps:

1. Manually remove the entry for the current cluster from: `/opt/mapr/conf/mapr-clusters.conf`
2. Run `configure.sh`.

16386: If you enable centralized logging on a cluster that was using YARN log aggregation in Version 4.0.1 prior to upgrading to version 4.1.0, you can no longer access previously aggregated MapReduce logs from the HistoryServer UI.

Workaround: Perform the following steps to view previously aggregated MapReduce logs from the History Server UI:

- Use the `yarn logs` command to retrieve the logs for each MapReduce application. The output of this command contains `stdout`, `stderr`, `syslog` with specific delimiters.
- Parse the output of `yarn logs` command to create `syslog`, `stdout`, `stderr` files using UNIX tools such as `sed` or `awk`.
- Add the `syslog`, `stdout`, `stderr` files to the centralized logging directory with the following directory hierarchy:
`/var/mapr/local/<NodeManager node>/logs/yarn/userlogs/application_<applicationID>/container_<containerID>/`

You will need to create the application and container directories and provide the user that submitted the application the proper permissions on the files and directories.

For example, if `usera` submitted the application, `usera` should have the following permissions on the directories and log files:

- Application directories:
`drwxr-s--- 5 usera mapr 4096 2015-01-07 11:32 /var/mapr/local/qa-node101.qa.lab/logs/yarn/userlogs/application_<id>`
- Container directories:
`drwxr-s--- - usera mapr 3 2015-01-07 11:32 /var/mapr/local/qa-node101.qa.lab/logs/yarn/userlogs/application_<id>/container_<id>`
- Log files:
`-rw-r----- 2 usera mapr 290 2015-01-07 11:32 /var/mapr/local/qa-node101.qa.lab/logs/yarn/userlogs/application_<id>/container_<id>/stderr`

Note: After you complete the workaround, you will also be able to run `maprccli job linklogs` on these logs.

Resolved Issues

The following issues are resolved in Version 5.0.

Installation and Configuration

14379: Running `configure.sh` no longer decreases the `ulimit` value in `/etc/security/limits.conf`. In previous releases, `configure.sh` had been decreasing the value in the line `mapr - nofile 64000` in the `limits.conf` file.

17815: When `isDB=false` in `warden.conf`, `configure.sh -R` no longer sets the `mfs.heapsize.percent` to 35 in `warden.conf`.

18382: Non-standard SSH ports are now supported in rolling upgrades and patch installations. Use the `--ssh_port` option.

19084: The `find` command in `/opt/mapr/server/configure-common.sh` was updated to find files via symbolic links.

Metrics Database

17541: The MapR Metrics database is now certified to work with MariaDB/InnoDB.

MapR-DB

18191: The CLDB no longer crashes when the following three conditions are true:

- A MapR-DB client Java application issues a scan or get request that uses FilterList.
- The application creates the FilterList object without passing in the required parameters.

For example, a best practice for creating a FilterList object is to use code such as this:

```
List<Filter> filters = new ArrayList<Filter>();
filters.add(filter1);
filters.add(filter2);

FilterList filterList = new FilterList(FilterList.Operator.MUST_PASS_ONE, filters);
scan.setFilter(filterList);
```

However, the next code sample shows code that matches this second error condition and does not pass in the required parameters:

```
FilterList filterList = new FilterList();
scan.setFilter(filterList);
```

- At least one access control expression has been defined for a column family in the table.

17397: YCSB performance was degraded between patches on MapR 3.1.1. A new patch was built to solve the problem.

18108: Truncating MapR-DB tables no longer resets access control expressions on the tables to u:<table creator>.

18567: You can now use the HBase shell or an HBase client to change the compression type of a column family in a MapR-DB table.

19008: Modifications that are made to a table's schema by means of the MapR-DB C APIs are now immediately available to the next operation on the table by a C API. In the 4.1 release, such modifications were available only after an interval of up to 5 minutes.

19342: A memory leak in `mdb_connection_create` was fixed.

MapR-FS

13092: To fix timeout issues, improvements were made to rename and unlink operations against MapR-FS.

18417: After an upgrade to 4.0.x, mirroring from a 3.x cluster to the upgraded cluster failed with a `VerifyReplicaError`. This issue is resolved in Version 5.0.

18636: NFS and `hadoop fs` commands now have the same permission requirements.

18791: A trailing `"/` character in the filepath of a `mkdir` command is now handled correctly when the command is submitted from MapR-FS.

YARN

15453: The `getDefaultProperties` method for the `SaslPropertiesResolver` class now returns `Map<String,String>` instead of `Map<String,Object>`.

15864: The ApplicationMaster UI works when the ApplicationMaster does not run on the same node as the ResourceManager.

16557: In Hadoop 2.5, the YARN fair scheduler became deadlocked when multiple jobs were started concurrently. In MapR Version 5.0, which supports Hadoop 2.7, the queue configuration parameter `queueMaxAMShareDefault` is set to 0.5f, which solves this problem.

17652: Jobs no longer fail when you specify the mapper for a streaming job as a binary without its full path.

17811: A problem with the YARN FIFO Scheduler was fixed as part of the upgrade to Hadoop 2.7.

17964, 17965, 17966: The following YARN issues are fixed in Version 5.0: [YARN-2273](#),

[YARN-2313](#), and [YARN-2874](#).

17983: When YARN log aggregation was enabled, only the `mapr` user could view job logs; in Version 5.0, regular users can view the logs generated by the jobs they run.

18032: Excessive logging by `org.apache.hadoop.util` is fixed by implementing a Hadoop patch that changes the log level to debug.

18364: The built-in Jetty server for both the YARN resource manager UI and the Job History server now accepts header buffers up to 64 KB in size when `https` is enabled.

18498: The health of the local volume and sub-directories is periodically checked and repaired (for example, if there is a disk failure that brings down the local volume) by restoring missing directories and volumes.

18900: The NodeManager no longer blocks new applications from starting while cleaning up old applications.

Scheduling

18474: MapR now installs an empty fair-scheduler.xml file that you can use to refresh queue settings.

CLI Commands

18637: The maprci volume rename command is now working correctly; the maprci volume info command no longer returns both the old and new volume names.

CLDB

17755: Improvements are made to the replication manager to improve container list iterator and log additional information about replication queues. Additionally, implemented as a patch for 4.x.

18020: Reformatting of disks and addition of new storage pools no longer causes the master CLDB to shut down with KvStoreException errors.

MapReduce

15020: TaskScheduler methods were fixed so that the Hadoop on Mesos project could be compiled against the MapR hadoop1 jar.

18253: Jobs no longer fail when processing files with a large input splits.

18525: Mapper tasks were considered to have completed successfully although they failed to create outputs. When tasks do not create outputs in Version 5.0, they will fail on the affected node, but they may be rescheduled on another node for execution.

POSIX Client

18662, 18684: The mapr-loopbacknfs initialization script no longer fails to run after a system reboot.

18663: The /usr/local/mapr-loopbacknfs/mapr-clusters.conf file is now updated correctly by the initscript: /usr/local/mapr-loopbacknfs/initscripts/mapr-loopbacknfs

18667: The maprloopback-nfs package for SUSE Linux platforms no longer depends on lsb packages.

NFS

18017: md5sum calculations now return consistent results across multiple NFS clients.

18489: NFS export entry line size is increased from 2048 to 8192 characters.

18490: Users no longer see permission denied errors via NFS when their UID is a member of more than 16 groups. When the user is added to a new group, an administrator should run the `maprci nfsmgmt refreshgidcache` command, or the user can wait an hour to access files that are available to the new group.

Ecosystem

16811: When Pig uses HCatLoader on a secure MapReduce v1 Kerberos cluster, jobs no longer fail with the following error: Provider org.apache.hadoop.hdfs.HttpFileSystem\$TokenManager not found

18028: A Teradata QueryGrid application integrated with MapR no longer fails with an exception when processing large amounts of data on Teradata nodes.

18174: Oozie jobs no longer fail with the following exception when Zero Configuration Failover is configured for the ResourceManager:
"java.lang.RuntimeException: Unable to determine ResourceManager service address from Zookeeper at localhost:5181"

18769: HBase 0.98.9 replication no longer fails.

Alarms

16967: You can now specify a location for storing core files by using the standard Linux `core_pattern` file.

MCS

19003: Test emails can now be sent via the MCS after configuring SMTP.

License Server

18503: License generation server is fixed to generate valid expiry dates for the POSIX license.