

# Hadoop Commands

All Hadoop commands are invoked by the `bin/hadoop` script.

When you run these commands, you can specify the MapReduce mode in two different ways:

1. Use the `hadoop` keyword and specify the mode explicitly, where `classic` mode refers to Hadoop 1.x and `yarn` mode refers to Hadoop 2.x.
2. Use the `hadoop1` or `hadoop2` keyword and do not specify the mode.

For example, the following commands are equivalent:

```
root@testnode100:/opt/mapr/conf/conf.d# hadoop2 conf | grep mapreduce.map.memory.mb
<property><name>mapreduce.map.memory.mb</name><value>1024</value></property>

root@testnode100:/opt/mapr/conf/conf.d# hadoop -yarn conf | grep
mapreduce.map.memory.mb
<property><name>mapreduce.map.memory.mb</name><value>1024</value></property>
```

## Syntax Summary

The following syntax summary applies to all commands:

```
hadoop [-yarn|-classic] [--config confdir] [COMMAND] [GENERIC_OPTIONS] [COMMAND_OPTIONS]
```

```
hadoop1 [--config confdir] [COMMAND] [GENERIC_OPTIONS] [COMMAND_OPTIONS]
```

```
hadoop2 [--config confdir] [COMMAND] [GENERIC_OPTIONS] [COMMAND_OPTIONS]
```

Hadoop has an option parsing framework that employs parsing generic options as well as running classes.

COMMAND_OPTION	Description
<code>-mode</code>	Specifies the Hadoop version: <code>yarn</code> or <code>classic</code>  Alternatively, you can use a <code>hadoop1</code> or <code>hadoop2</code> command without setting the mode.  If you use a <code>hadoop</code> command (instead of <code>hadoop1</code> or <code>hadoop2</code> ) and do not set the mode, the command runs in the mode set by the <code>MAPR_MAPREDUCE_MODE</code> environment variable.  If this variable is not set, the command runs in the mode set in the Hadoop version file on the node ( <code>default_mode = yarn</code> or <code>classic</code> ).
<code>--config confdir</code>	Overwrites the default Configuration directory. Default is <code>\${HADOOP_HOME}/conf</code> .
COMMAND	Various commands with their options are described in the following sections.
GENERIC_OPTIONS	The common set of options supported by multiple commands.
COMMAND_OPTIONS	Various command options are described in the following sections.



### Useful Information

Running the `hadoop` script without any arguments prints the help description for all commands.

## Supported Commands for Hadoop 1.x

MapR supports the following `hadoop` commands for Hadoop 1.x:

Command	Description
---------	-------------

<code>archive</code> <code>-archiveName</code> <code>NAME &lt;src&gt;*</code> <code>&lt;dest&gt;</code>	The <code>hadoop archive</code> command creates a Hadoop archive, a file that contains other files. A Hadoop archive always has a <code>.har</code> extension.
<code>classpath</code>	The <code>hadoop classpath</code> command prints the class path needed to access the Hadoop JAR and the required libraries.
<code>conf</code>	The <code>hadoop conf</code> command prints the configuration information for the current node.
<code>daemonlog</code>	The <code>hadoop daemonlog</code> command may be used to get or set the log level of Hadoop daemons.
<code>distcp</code> <code>&lt;source&gt;</code> <code>&lt;destination&gt;</code>	The <code>hadoop distcp</code> command is a tool for large inter- and intra-cluster copying. It uses MapReduce to effect its distribution, error handling and recovery, and reporting. It expands a list of files and directories into input to map tasks, each of which will copy a partition of the files specified in the source list.
<code>fs</code>	The <code>hadoop fs</code> command runs a generic filesystem user client that interacts with the MapR filesystem (MapR-FS).
<code>jar &lt;jar&gt;</code>	The <code>hadoop jar</code> command runs a JAR file. Users can bundle their MapReduce code in a JAR file and execute it using this command.
<code>job</code>	Manipulates MapReduce jobs.
<code>mfs</code>	The <code>hadoop mfs</code> command performs operations on directories in the cluster. The main purposes of <code>hadoop mfs</code> are to display directory information and contents, to create symbolic links, and to set compression and chunk size on a directory.
<code>mradmin</code>	Runs a MapReduce admin client.
<code>pipes</code>	Runs a pipes job.
<code>queue</code>	Gets information about job queues.
<code>version</code>	The <code>hadoop version</code> command prints the Hadoop software version.

## Supported Commands for Hadoop 2.x

MapR supports the following `hadoop` commands for Hadoop 2.x:

Command	Description
<code>archive</code> <code>-archiveName</code> <code>NAME &lt;src&gt;*</code> <code>&lt;dest&gt;</code>	Creates a Hadoop archive, a file that contains other files. A Hadoop archive always has a <code>.har</code> extension.
<code>CLASSNAME</code>	The <code>hadoop</code> script can be used to invoke any class.  <code>hadoop CLASSNAME</code> runs the class named <code>CLASSNAME</code> .
<code>classpath</code>	Prints the class path needed to access the Hadoop JAR and the required libraries.
<code>conf</code>	The <code>hadoop conf</code> command prints the configuration information for the current node.
<code>daemonlog</code>	The <code>hadoop daemonlog</code> command may be used to get or set the log level of Hadoop daemons.
<code>distcp</code> <code>&lt;source&gt;</code> <code>&lt;destination&gt;</code>	The <code>hadoop distcp</code> command is a tool for large inter- and intra-cluster copying. It uses MapReduce to effect its distribution, error handling and recovery, and reporting. It expands a list of files and directories into input to map tasks, each of which will copy a partition of the files specified in the source list.
<code>fs</code>	The <code>hadoop fs</code> command runs a generic filesystem user client that interacts with the MapR filesystem (MapR-FS).
<code>jar &lt;jar&gt;</code>	The <code>hadoop jar</code> command runs a JAR file. Users can bundle their MapReduce code in a JAR file and execute it using this command.
<code>mfs</code>	The <code>hadoop mfs</code> command performs operations on directories in the cluster. The main purposes of <code>hadoop mfs</code> are to display directory information and contents, to create symbolic links, and to set compression and chunk size on a directory.
<code>version</code>	The <code>hadoop version</code> command prints the Hadoop software version.



For Hadoop2, some `hadoop` commands are deprecated and replaced by the `mapred` command.

For example, if you run the `hadoop job` command, you see this message:

```
# hadoop job
DEPRECATED: Use of this script to execute mapred command is deprecated.
Instead, use the mapred command for it.
```

The syntax for the `mapred` command is:

```
mapred [--config confdir] COMMAND
```

Commands used with `mapred` include:

Command	Description
<code>historyserver</code>	Runs job history servers as a standalone daemon
<code>hsadmin</code>	The job history server admin interface
<code>job</code>	Manipulates MapReduce jobs
<code>pipes</code>	Runs a <code>pipes</code> job
<code>queue</code>	Gets information regarding <code>JobQueues</code>

## Unsupported Commands

MapR does not support the following Hadoop commands:

- `balancer`
- `datanode`
- `dfsadmin`
- `fsck`
- `fetchdt`
- `jobtracker`
- `namenode`
- `secondarynamenode`
- `tasktracker`

## Generic Options

Implement the [Tool](#) interface to make the following command-line options available for many of the Hadoop commands.

The following generic options are supported by the `distcp`, `fs`, `job`, `mradmin`, `pipes`, and `queue` Hadoop commands:

Generic Option	Description
<code>-conf &lt;filename1 filename2 ...&gt;</code>	Add the specified configuration files to the list of resources available in the configuration.
<code>-D &lt;property=value&gt;</code>	Set a value for the specified Hadoop configuration property.
<code>-fs &lt;local filesystem URI&gt;</code>	Set the URI of the default filesystem.
<code>-jt &lt;local jobtracker:port&gt;</code>	Specify a jobtracker for a given host and port. This command option is a shortcut for <code>-D mapred.job.tracker=host:port</code>
<code>-files &lt;file1,file2,...&gt;</code>	Specify files to be copied to the map reduce cluster.
<code>-libjars &lt;jar1,jar2,...&gt;</code>	Specify JAR files to be included in the classpath of the mapper and reducer tasks.

-archives  
<archive1,archive2,...>

Specify archive files (JAR, tar, tar.gz, ZIP) to be copied and unarchived on the task node.