

HDFS Commands Guide

- [Overview](#)
- [User Commands](#)
 - [classpath](#)
 - [dfs](#)
 - [fetchdt](#)
 - [fsck](#)
 - [getconf](#)
 - [groups](#)
 - [lsSnapshottableDir](#)
 - [jmxget](#)
 - [oiv](#)
 - [oiv_legacy](#)
 - [snapshotDiff](#)
 - [version](#)
- [Administration Commands](#)
 - [balancer](#)
 - [cacheadmin](#)
 - [crypto](#)
 - [datanode](#)
 - [dfsadmin](#)
 - [haadmin](#)
 - [journalnode](#)
 - [mover](#)
 - [namenode](#)
 - [nfs3](#)
 - [portmap](#)
 - [secondarynamenode](#)
 - [storagepolicies](#)
 - [zkfc](#)
- [Debug Commands](#)
 - [verifyMeta](#)
 - [computeMeta](#)
 - [recoverLease](#)

Overview

All HDFS commands are invoked by the `bin/hdfs` script. Running the `hdfs` script without any arguments prints the description for all commands.

Usage: `hdfs [SHELL_OPTIONS] COMMAND [GENERIC_OPTIONS] [COMMAND_OPTIONS]`

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

COMMAND_OPTIONS	Description
<code>--config</code> <code>--loglevel</code>	The common set of shell options. These are documented on the Commands Manual page.
GENERIC_OPTIONS	The common set of options supported by multiple commands. See the Hadoop Commands Manual for more information.
COMMAND COMMAND_OPTIONS	Various commands with their options are described in the following sections. The commands have been grouped into User Commands and Administration Commands .

User Commands

Commands useful for users of a hadoop cluster.

classpath

Usage: `hdfs classpath [--glob | --jar <path> | -h | --help]`

COMMAND_OPTION	Description
<code>--glob</code>	expand wildcards
<code>--jar path</code>	write classpath as manifest in jar named <i>path</i>
<code>-h, --help</code>	print help

Prints the class path needed to get the Hadoop jar and the required libraries. If called without arguments, then prints the classpath set up by the command scripts, which is likely to contain wildcards in the classpath entries. Additional options print the classpath after wildcard expansion or write the classpath into the manifest of a jar file. The latter is useful in environments where wildcards cannot be used and the expanded classpath exceeds the maximum supported command line length.

dfs

Usage: `hdfs dfs [COMMAND [COMMAND_OPTIONS]]`

Run a filesystem command on the file system supported in Hadoop. The various `COMMAND_OPTIONS` can be found at [File System Shell Guide](#).

fetchdt

Usage: `hdfs fetchdt <opts> <token_file_path>`

COMMAND_OPTION	Description
<code>--webservice NN_Url</code>	Url to contact NN on (starts with http or https)
<code>--renewer name</code>	Name of the delegation token renewer
<code>--cancel</code>	Cancel the delegation token
<code>--renew</code>	Renew the delegation token. Delegation token must have been fetched using the <code>--renewer name</code> option.
<code>--print</code>	Print the delegation token
<code>token_file_path</code>	File path to store the token into.

Gets Delegation Token from a NameNode. See [fetchdt](#) for more info.

fsck

Usage:

```
hdfs fsck <path>
    [-list-corruptfileblocks |
    [-move | -delete | -openforwrite]
    [-files [-blocks [-locations | -racks | -replicaDetails | -upgrad
    [-includeSnapshots]
    [-storagepolicies] [-blockId <blk_Id>]
```

COMMAND_OPTION	Description
----------------	-------------

COMMAND_OPTION	Description
<i>path</i>	Start checking from this path.
-delete	Delete corrupted files.
-files	Print out files being checked.
-files -blocks	Print out the block report
-files -blocks -locations	Print out locations for every block.
-files -blocks -racks	Print out network topology for data-node locations.
-files -blocks -replicaDetails	Print out each replica details.
-files -blocks -upgradedomains	Print out upgrade domains for every block.
-includeSnapshots	Include snapshot data if the given path indicates a snapshottable directory or there are snapshottable directories under it.
-list-corruptfileblocks	Print out list of missing blocks and files they belong to.
-move	Move corrupted files to /lost+found.
-openforwrite	Print out files opened for write.
-storagepolicies	Print out storage policy summary for the blocks.
-blockId	Print out information about the block.

Runs the HDFS filesystem checking utility. See [fsck](#) for more info.

getconf

Usage:

```
hdfs getconf -namenodes
hdfs getconf -secondaryNameNodes
hdfs getconf -backupNodes
hdfs getconf -includeFile
hdfs getconf -excludeFile
hdfs getconf -nnRpcAddresses
hdfs getconf -confKey [key]
```

COMMAND_OPTION	Description
-namenodes	gets list of namenodes in the cluster.
-secondaryNameNodes	gets list of secondary namenodes in the cluster.
-backupNodes	gets list of backup nodes in the cluster.
-includeFile	gets the include file path that defines the datanodes that can join the cluster.
-excludeFile	gets the exclude file path that defines the datanodes that need to decommissioned.
-nnRpcAddresses	gets the namenode rpc addresses
-confKey [key]	gets a specific key from the configuration

Gets configuration information from the configuration directory, post-processing.

groups

Usage: `hdfs groups [username ...]`

Returns the group information given one or more usernames.

lsSnapshottableDir

Usage: `hdfs lsSnapshottableDir [-help]`

COMMAND_OPTION

`-help`

Description

print help

Get the list of snapshottable directories. When this is run as a super user, it returns all snapshottable directories. Otherwise it returns those directories that are owned by the current user.

jmxget

Usage: `hdfs jmxget [-localVM ConnectorURL | -port port | -server mbeanserver | -service service]`

COMMAND_OPTION

Description

`-help`

print help

`-localVM ConnectorURL`

connect to the VM on the same machine

`-port mbean server port`

specify mbean server port, if missing it will try to connect to MBean Server in the same VM

`-service`

specify jmx service, either DataNode or NameNode, the default

Dump JMX information from a service.

oev

Usage: `hdfs oev [OPTIONS] -i INPUT_FILE -o OUTPUT_FILE`

Required command line arguments:

COMMAND_OPTION

Description

`-i,--inputFile arg`

edits file to process, xml (case insensitive) extension means XML format, any other filename means binary format

`-o,--outputFile arg`

Name of output file. If the specified file exists, it will be overwritten, format of the file is determined by -p option

Optional command line arguments:

COMMAND_OPTION

Description

`-f,--fix-txids`

Renumber the transaction IDs in the input, so that there are no gaps or invalid transaction IDs.

`-h,--help`

Display usage information and exit

`-r,--recover`

When reading binary edit logs, use recovery mode. This will give you the chance to skip corrupt parts of the edit log.

COMMAND_OPTION Description

<code>-p,--processor <i>arg</i></code>	Select which type of processor to apply against image file, currently supported processors are: binary (native binary format that Hadoop uses), xml (default, XML format), stats (prints statistics about edits file)
<code>-v,--verbose</code>	More verbose output, prints the input and output filenames, for processors that write to a file, also output to screen. On large image files this will dramatically increase processing time (default is false).

Hadoop offline edits viewer.

oiv

Usage: `hdfs oiv [OPTIONS] -i INPUT_FILE`

Required command line arguments:

COMMAND_OPTION Description

<code>-i,--inputFile <i>arg</i></code>	edits file to process, xml (case insensitive) extension means XML format, any other filename means binary format
----------------------------------------	------------------------------------------------------------------------------------------------------------------

Optional command line arguments:

COMMAND_OPTION Description

<code>-h,--help</code>	Display usage information and exit
<code>-o,--outputFile <i>arg</i></code>	Name of output file. If the specified file exists, it will be overwritten, format of the file is determined by -p option
<code>-p,--processor <i>arg</i></code>	Select which type of processor to apply against image file, currently supported processors are: binary (native binary format that Hadoop uses), xml (default, XML format), stats (prints statistics about edits file)

Hadoop Offline Image Viewer for newer image files.

oiv_legacy

Usage: `hdfs oiv_legacy [OPTIONS] -i INPUT_FILE -o OUTPUT_FILE`

COMMAND_OPTION Description

<code>-h,--help</code>	Display usage information and exit
<code>-i,--inputFile <i>arg</i></code>	edits file to process, xml (case insensitive) extension means XML format, any other filename means binary format
<code>-o,--outputFile <i>arg</i></code>	Name of output file. If the specified file exists, it will be overwritten, format of the file is determined by -p option

Hadoop offline image viewer for older versions of Hadoop.

snapshotDiff

Usage: `hdfs snapshotDiff <path> <fromSnapshot> <toSnapshot>`

Determine the difference between HDFS snapshots. See the [HDFS Snapshot Documentation](#) for more information.

version

Usage: `hdfs version`

Prints the version.

Administration Commands

Commands useful for administrators of a hadoop cluster.

balancer

Usage:

```
hdfs balancer
  [-threshold <threshold>]
  [-policy <policy>]
  [-exclude [-f <hosts-file> | <comma-separated list of hosts>]]
  [-include [-f <hosts-file> | <comma-separated list of hosts>]]
  [-source [-f <hosts-file> | <comma-separated list of hosts>]]
  [-blockpools <comma-separated list of blockpool ids>]
  [-idleiterations <idleiterations>]
```

COMMAND_OPTION

Description

<code>-policy <policy></code>	<code>datanode</code> (default): Cluster is balanced if each datanode is balanced. <code>blockpool</code> : Cluster is balanced if each block pool in each datanode is balanced.
<code>-threshold <threshold></code>	Percentage of disk capacity. This overwrites the default threshold.
<code>-exclude -f <hosts-file> <comma-separated list of hosts></code>	Excludes the specified datanodes from being balanced by the balancer.
<code>-include -f <hosts-file> <comma-separated list of hosts></code>	Includes only the specified datanodes to be balanced by the balancer.
<code>-source -f <hosts-file> <comma-separated list of hosts></code>	Pick only the specified datanodes as source nodes.
<code>-blockpools <comma-separated list of blockpool ids></code>	The balancer will only run on blockpools included in this list.
<code>-idleiterations <iterations></code>	Maximum number of idle iterations before exit. This overwrites the default <code>idleiterations(5)</code> .

Runs a cluster balancing utility. An administrator can simply press Ctrl-C to stop the rebalancing process. See [Balancer](#) for more details.

Note that the `blockpool` policy is more strict than the `datanode` policy.

Besides the above command options, a pinning feature is introduced starting from 2.7.0 to prevent certain replicas from getting moved by balancer/mover. This pinning feature is disabled by default, and can be enabled by configuration property `"dfs.datanode.block-pinning.enabled"`. When enabled, this feature only affects blocks that are written to favored nodes specified in the `create()` call. This feature is useful when we want to maintain the data locality, for applications such as HBase regionserver.

cacheadmin

Usage: `hdfs cacheadmin -addDirective -path <path> -pool <pool-name> [-force] [-replication <replication>] [-ttl <time-to-live>]`

See the [HDFS Cache Administration Documentation](#) for more information.

crypto

Usage:

```
hdfs crypto -createZone -keyName <keyName> -path <path>
hdfs crypto -listZones
hdfs crypto -provisionTrash -path <path>
hdfs crypto -help <command-name>
```

See the [HDFS Transparent Encryption Documentation](#) for more information.

datanode

Usage: `hdfs datanode [-regular | -rollback | -rollingupgrade rollback]`

COMMAND_OPTION	Description
-regular	Normal datanode startup (default).
-rollback	Rollback the datanode to the previous version. This should be used after stopping the datanode and distributing the old hadoop version.
-rollingupgrade rollback	Rollback a rolling upgrade operation.

Runs a HDFS datanode.

dfsadmin

Usage:

```
hdfs dfsadmin [GENERIC_OPTIONS]
    [-report [-live] [-dead] [-decommissioning]]
    [-safemode enter | leave | get | wait | forceExit]
    [-saveNamespace]
    [-rollEdits]
    [-restoreFailedStorage true | false | check]
    [-refreshNodes]
    [-setQuota <quota> <dirname>...<dirname>]
    [-clrQuota <dirname>...<dirname>]
    [-setSpaceQuota <quota> [-storageType <storagetype>] <dirname>...]
    [-clrSpaceQuota [-storageType <storagetype>] <dirname>...<dirname>]
    [-finalizeUpgrade]
    [-rollingUpgrade [<query> |<prepare> |<finalize>]]
    [-metasave filename]
    [-refreshServiceAcl]
```

```
[--refreshUserToGroupsMappings]
[--refreshSuperUserGroupsConfiguration]
[--refreshCallQueue]
[--refresh <host:ipc_port> <key> [arg1..argn]]
[--reconfig <datanode |...> <host:ipc_port> <start |status>]
[--printTopology]
[--refreshNamenodes datanodehost:port]
[--deleteBlockPool datanode-host:port blockpoolId [force]]
[--setBalancerBandwidth <bandwidth in bytes per second>]
[--getBalancerBandwidth <datanode_host:ipc_port>]
[--allowSnapshot <snapshotDir>]
[--disallowSnapshot <snapshotDir>]
[--fetchImage <local directory>]
[--shutdownDatanode <datanode_host:ipc_port> [upgrade]]
[--getDatanodeInfo <datanode_host:ipc_port>]
[--evictWriters <datanode_host:ipc_port>]
[--triggerBlockReport [--incremental] <datanode_host:ipc_port>]
[--help [cmd]]
```

COMMAND_OPTION

`--report [--live] [--dead]`
`--decommissioning`

Description

Reports basic filesystem information and statistics, The dfs usage can be different from “du” usage, because it measures raw space used by replication, checksums, snapshots and etc. on all the DNs. Optional flags may be used to filter the list of displayed DataNodes.

`--safemode enter|leave|get|wait|forceExit`

Safe mode maintenance command. Safe mode is a Namenode state in which it

1. does not accept changes to the name space (read-only)
2. does not replicate or delete blocks.

Safe mode is entered automatically at Namenode startup, and leaves safe mode automatically when the configured minimum percentage of blocks satisfies the minimum replication condition. If Namenode detects any anomaly then it will linger in safe mode till that issue is resolved. If that anomaly is the consequence of a deliberate action, then administrator can use `--safemode forceExit` to exit safe mode. The cases where `forceExit` may be required are

1. Namenode metadata is not consistent. If Namenode detects that metadata has been modified out of band and can cause data loss, then Namenode will enter `forceExit` state. At that point user can either restart Namenode with correct metadata files or `forceExit` (if data loss is acceptable).
2. Rollback causes metadata to be replaced and rarely it can trigger safe mode `forceExit` state in Namenode. In that case you may proceed by issuing `--safemode forceExit`.

Safe mode can also be entered manually, but then it can only be turned off manually as well.

`--saveNamespace`

Save current namespace into storage directories and reset edits log. Requires safe mode.

`--rollEdits`

Rolls the edit log on the active NameNode.

`--restoreFailedStorage true|false|check`

This option will turn on/off automatic attempt to restore failed storage replicas. If a failed storage becomes available again the system will attempt to restore edits and/or fsimage during checkpoint. ‘check’ option will return current setting.

`--refreshNodes`

Re-read the hosts and exclude files to update the set of Datanodes that are allowed to connect to the Namenode and those that should be decommissioned or recommissioned.

`--setQuota <quota> <dirname>...<dirname>`

See [HDFS Quotas Guide](#) for the detail.

`--clrQuota <dirname>...<dirname>`

See [HDFS Quotas Guide](#) for the detail.

`--setSpaceQuota <quota> [--storageType <storagetype>] <dirname>...<dirname>`

See [HDFS Quotas Guide](#) for the detail.

COMMAND_OPTION

```
-clrSpaceQuota [-storageType
<storageType>] <dirname>...<dirname>

-finalizeUpgrade

-rollingUpgrade [<query>|<prepare>|
<finalize>]

-metasave filename

-refreshServiceAcl

-refreshUserToGroupsMappings

-refreshSuperUserGroupsConfiguration

-refreshCallQueue

-refresh <host:ipc_port> <key>
[arg1..argn]

-reconfig <datanode |...> <host:ipc_port>
<start|status>

-printTopology

-refreshNamenodes datanodehost:port

-deleteBlockPool datanode-host:port
blockpoolId [force]

-setBalancerBandwidth <bandwidth in bytes
per second>

-getBalancerBandwidth
<datanode_host:ipc_port>

-allowSnapshot <snapshotDir>

-disallowSnapshot <snapshotDir>

-fetchImage <local directory>

-shutdownDatanode
<datanode_host:ipc_port> [upgrade]

-evictWriters <datanode_host:ipc_port>

-getDatanodeInfo
<datanode_host:ipc_port>
```

Description

See [HDFS Quotas Guide](#) for the detail.

Finalize upgrade of HDFS. Datanodes delete their previous version working directories, followed by Namenode doing the same. This completes the upgrade process.

See [Rolling Upgrade document](#) for the detail.

Save Namenode's primary data structures to *filename* in the directory specified by `hadoop.log.dir` property. *filename* is overwritten if it exists. *filename* will contain one line for each of the following

1. Datanodes heart beating with Namenode
2. Blocks waiting to be replicated
3. Blocks currently being replicated
4. Blocks waiting to be deleted

Reload the service-level authorization policy file.

Refresh user-to-groups mappings.

Refresh superuser proxy groups mappings

Reload the call queue from config.

Triggers a runtime-refresh of the resource specified by `<key>` on `<host:ipc_port>`. All other args after are sent to the host.

Start reconfiguration or get the status of an ongoing reconfiguration. The second parameter specifies the node type. Currently, only reloading DataNode's configuration is supported.

Print a tree of the racks and their nodes as reported by the Namenode

For the given datanode, reloads the configuration files, stops serving the removed block-pools and starts serving new block-pools.

If force is passed, block pool directory for the given blockpool id on the given datanode is deleted along with its contents, otherwise the directory is deleted only if it is empty. The command will fail if datanode is still serving the block pool. Refer to `refreshNamenodes` to shutdown a block pool service on a datanode.

Changes the network bandwidth used by each datanode during HDFS block balancing. `<bandwidth>` is the maximum number of bytes per second that will be used by each datanode. This value overrides the `dfs.balance.bandwidthPerSec` parameter. NOTE: The new value is not persistent on the DataNode.

Get the network bandwidth(in bytes per second) for the given datanode. This is the maximum network bandwidth used by the datanode during HDFS block balancing.

Allowing snapshots of a directory to be created. If the operation completes successfully, the directory becomes snapshottable. See the [HDFS Snapshot Documentation](#) for more information.

Disallowing snapshots of a directory to be created. All snapshots of the directory must be deleted before disallowing snapshots. See the [HDFS Snapshot Documentation](#) for more information.

Downloads the most recent fsimage from the NameNode and saves it in the specified local directory.

Submit a shutdown request for the given datanode. See [Rolling Upgrade document](#) for the detail.

Make the datanode evict all clients that are writing a block. This is useful if decommissioning is hung due to slow writers.

Get the information about the given datanode. See [Rolling Upgrade document](#) for the detail.

COMMAND_OPTION

```
-triggerBlockReport [-incremental]
<datanode_host:ipc_port>
-help [cmd]
```

Description

Trigger a block report for the given datanode. If 'incremental' is specified, it will be otherwise, it will be a full block report.

Displays help for the given command or all commands if none is specified.

Runs a HDFS dfsadmin client.

haadmin

Usage:

```
hdfs haadmin -checkHealth <serviceId>
hdfs haadmin -failover [--forcefence] [--forceactive] <serviceId> <serv
hdfs haadmin -getServiceState <serviceId>
hdfs haadmin -help <command>
hdfs haadmin -transitionToActive <serviceId> [--forceactive]
hdfs haadmin -transitionToStandby <serviceId>
```

COMMAND_OPTION

Description

```
-checkHealth      check the health of the given NameNode
-failover         initiate a failover between two NameNodes
-getServiceState  determine whether the given NameNode is Active or Standby
-transitionToActive transition the state of the given NameNode to Active (Warning: No fencing is done)
-transitionToStandby transition the state of the given NameNode to Standby (Warning: No fencing is done)
```

See [HDFS HA with NFS](#) or [HDFS HA with QJM](#) for more information on this command.

journalnode

Usage: hdfs journalnode

This comamnd starts a journalnode for use with [HDFS HA with QJM](#).

mover

Usage: `hdfs mover [-p <files/dirs> | -f <local file name>]`

COMMAND_OPTION	Description
<code>-f <local file></code>	Specify a local file containing a list of HDFS files/dirs to migrate.
<code>-p <files/dirs></code>	Specify a space separated list of HDFS files/dirs to migrate.

Runs the data migration utility. See [Mover](#) for more details.

Note that, when both `-p` and `-f` options are omitted, the default path is the root directory.

In addition, a pinning feature is introduced starting from 2.7.0 to prevent certain replicas from getting moved by balancer/mover. This pinning feature is disabled by default, and can be enabled by configuration property `"dfs.datanode.block-pinning.enabled"`. When enabled, this feature only affects blocks that are written to favored nodes specified in the `create()` call. This feature is useful when we want to maintain the data locality, for applications such as HBase regionserver.

namenode

Usage:

```
hdfs namenode [-backup] |
               [-checkpoint] |
               [-format [-clusterid cid] [-force] [-nonInteractive] ] |
               [-upgrade [-clusterid cid] [-renameReserved<k-v pairs>] ] |
               [-upgradeOnly [-clusterid cid] [-renameReserved<k-v pairs>] ] |
               [-rollback] |
               [-rollingUpgrade <rollback |started> ] |
               [-finalize] |
               [-importCheckpoint] |
               [-initializeSharedEdits] |
               [-bootstrapStandby [-force] [-nonInteractive] [-skipSharedEditsCl
               [-recover [-force] ] |
               [-metadataVersion ]
```

COMMAND_OPTION	Description
<code>-backup</code>	Start backup node.
<code>-checkpoint</code>	Start checkpoint node.
<code>-format [-clusterid cid]</code> <code>[-force] [-nonInteractive]</code>	Formats the specified NameNode. It starts the NameNode, formats it and then shut it down. <code>-force</code> option formats if the name directory exists. <code>-nonInteractive</code> option aborts if the name directory exists, unless <code>-force</code> option is specified.
<code>-upgrade [-clusterid cid]</code> <code>[-renameReserved <k-v pairs>]</code>	Namenode should be started with upgrade option after the distribution of new Hadoop version.
<code>-upgradeOnly [-clusterid cid]</code> <code>[-renameReserved <k-v pairs>]</code>	Upgrade the specified NameNode and then shutdown it.
<code>-rollback</code>	Rollback the NameNode to the previous version. This should be used after stopping the cluster and distributing the old Hadoop version.
<code>-rollingUpgrade</code> <code><downgrade rollback started></code>	See Rolling Upgrade document for the detail.
<code>-finalize</code>	No longer supported. Use <code>dfsadmin -finalizeUpgrade</code> instead.

COMMAND_OPTION	Description
<code>-importCheckpoint</code>	Loads image from a checkpoint directory and save it into the current one. Checkpoint dir is read from property <code>dfs.namenode.checkpoint.dir</code>
<code>-initializeSharedEdits</code>	Format a new shared edits dir and copy in enough edit log segments so that the standby NameNode can start up.
<code>-bootstrapStandby [-force] [-nonInteractive] [-skipSharedEditsCheck]</code>	Allows the standby NameNode's storage directories to be bootstrapped by copying the latest namespace snapshot from the active NameNode. This is used when first configuring an HA cluster. The option <code>-force</code> or <code>-nonInteractive</code> has the same meaning as that described in <code>namenode -format</code> command. <code>-skipSharedEditsCheck</code> option skips edits check which ensures that we have enough edits already in the shared directory to start up from the last checkpoint on the active.
<code>-recover [-force]</code>	Recover lost metadata on a corrupt filesystem. See HDFS User Guide for the detail.
<code>-metadataVersion</code>	Verify that configured directories exist, then print the metadata versions of the software and the image.

Runs the namenode. More info about the upgrade, rollback and finalize is at [Upgrade Rollback](#).

nfs3

Usage: `hdfs nfs3`

This comamnd starts the NFS3 gateway for use with the [HDFS NFS3 Service](#).

portmap

Usage: `hdfs portmap`

This comamnd starts the RPC portmap for use with the [HDFS NFS3 Service](#).

secondarynamenode

Usage: `hdfs secondarynamenode [-checkpoint [force]] | [-format] | [-geteditsize]`

COMMAND_OPTION	Description
<code>-checkpoint [force]</code>	Checkpoints the SecondaryNameNode if EditLog size \geq <code>fs.checkpoint.size</code> . If <code>force</code> is used, checkpoint irrespective of EditLog size.
<code>-format</code>	Format the local storage during startup.
<code>-geteditsize</code>	Prints the number of uncheckpointed transactions on the NameNode.

Runs the HDFS secondary namenode. See [Secondary Namenode](#) for more info.

storagepolicies

Usage: `hdfs storagepolicies`

Lists out all storage policies. See the [HDFS Storage Policy Documentation](#) for more information.

zkfc

Usage: `hdfs zkfc [-formatZK [-force] [-nonInteractive]]`

COMMAND_OPTION	Description
-formatZK	Format the Zookeeper instance. -force: formats the znode if the znode exists. -nonInteractive: formats the znode aborts if the znode exists, unless -force option is specified.
-h	Display help

This comamnd starts a Zookeeper Failover Controller process for use with [HDFS HA with QJM](#).

Debug Commands

Useful commands to help administrators debug HDFS issues. These commands are for advanced users only.

verifyMeta

Usage: `hdfs debug verifyMeta -meta <metadata-file> [-block <block-file>]`

COMMAND_OPTION	Description
-block <i>block-file</i>	Optional parameter to specify the absolute path for the block file on the local file system of the data node.
-meta <i>metadata-file</i>	Absolute path for the metadata file on the local file system of the data node.

Verify HDFS metadata and block files. If a block file is specified, we will verify that the checksums in the metadata file match the block file.

computeMeta

Usage: `hdfs debug computeMeta -block <block-file> -out <output-metadata-file>`

COMMAND_OPTION	Description
-block <i>block-file</i>	Absolute path for the block file on the local file system of the data node.
-out <i>output-metadata-file</i>	Absolute path for the output metadata file to store the checksum computation result from the block file.

Compute HDFS metadata from block files. If a block file is specified, we will compute the checksums from the block file, and save it to the specified output metadata file.

NOTE: Use at your own risk! If the block file is corrupt and you overwrite it's meta file, it will show up as 'good' in HDFS, but you can't read the data. Only use as a last measure, and when you are 100% certain the block file is good.

recoverLease

Usage: `hdfs debug recoverLease -path <path> [-retries <num-retries>]`

COMMAND_OPTION	Description
[-path <i>path</i>]	HDFS path for which to recover the lease.
[-retries <i>num-retries</i>]	Number of times the client will retry calling recoverLease. The default number of retries is 1.

Recover the lease on the specified path. The path must reside on an HDFS filesystem. The default number of retries is 1.