HADOOP SHELL COMMANDS

## help

Usage: hadoop fs –help [command]

Return usage output.

**Example**:

> hadoop fs -help ls

-ls [-d] [-h] [-R] [<path> ...] :

List the contents that match the specified file pattern. If path is not

specified, the contents of /user/<currentUser> will be listed. Directory entries

are of the form:

permissions - userId groupId sizeOfDirectory(in bytes)

modificationDate(yyyy-MM-dd HH:mm) directoryName

and file entries are of the form:

permissions numberOfReplicas userId groupId sizeOfFile(in bytes)

modificationDate(yyyy-MM-dd HH:mm) fileName

-d Directories are listed as plain files.

-h Formats the sizes of files in a human-readable fashion rather than a number

of bytes.

-R Recursively list the contents of directories.

pedgenode02:uzinf100:/ai/src/prod/global/sand/inf/inf\_new\_pvt

## ls

**Usage**: hdfs dfs -ls <args>

For a file returns stat on the file with the following format:

permissions number\_of\_replicas userid groupid filesize modification\_date modification\_time filename

For a directory it returns list of its direct children as in Unix. A directory is listed as:

permissions userid groupid modification\_date modification\_time dirname

Option:

With -d option, only displays the directories.

With -R option, display the output recursively through the directory structure.

**Example**:

> hadoop fs -ls /ai/data/prod/hdfs/inf/inf\_new\_pvt/main/

Found 4 items

drwxrwxrwt+ - uzinf100 abiuserp 0 2017-01-31 02:33 /ai/data/prod/hdfs/inf/inf\_new\_pvt/main/.WORK-serial

drwxrwxr-x+ - uzinf100 abiuserp 0 2017-01-14 08:05 /ai/data/prod/hdfs/inf/inf\_new\_pvt/main/profile.new

drwxrwxr-x+ - uzinf100 abiuserp 0 2016-12-03 03:31 /ai/data/prod/hdfs/inf/inf\_new\_pvt/main/source.new

drwxrwxr-x+ - uzinf100 abiuserp 0 2017-01-31 02:36 /ai/data/prod/hdfs/inf/inf\_new\_pvt/main/today.new

pedgenode02:uzinf100:/ai/src/prod/global/sand/inf/inf\_new\_pvt

> hadoop fs -ls -d /ai/data/prod/hdfs/inf/inf\_new\_pvt/main/today.new/today.new.t\_applt\*

drwxrwxr-x - uzinf100 abiuserp 0 2017-01-31 01:30 /ai/data/prod/hdfs/inf/inf\_new\_pvt/main/today.new/today.new.t\_applt\_id

drwxrwxr-x - uzinf100 abiuserp 0 2017-01-31 00:39 /ai/data/prod/hdfs/inf/inf\_new\_pvt/main/today.new/today.new.t\_applt\_to\_acct

pedgenode02:uzinf100:/ai/src/prod/global/sand/inf/inf\_new\_pvt

> hadoop fs -ls -R /ai/archive/prod/hdfs/inf/inf\_aj\_pvt/./daily\_2017-01-19\_032641/

drwxr-xr-x - uzinf100 abiuserp 0 2017-01-19 03:30 /ai/archive/prod/hdfs/inf/inf\_aj\_pvt/daily\_2017-01-19\_032641/avro\_schema

-rw-r--r-- 3 uzinf100 abiuserp 1652 2017-01-19 03:30 /ai/archive/prod/hdfs/inf/inf\_aj\_pvt/daily\_2017-01-19\_032641/avro\_schema/avro\_schema.aj.v\_contractors\_arriba.avsc

drwxr-xr-x - uzinf100 abiuserp 0 2017-01-19 03:30 /ai/archive/prod/hdfs/inf/inf\_aj\_pvt/daily\_2017-01-19\_032641/hadoop\_delta

drwxr-xr-x - uzinf100 abiuserp 0 2017-01-19 03:30 /ai/archive/prod/hdfs/inf/inf\_aj\_pvt/daily\_2017-01-19\_032641/registered\_source

drwxr-xr-x - uzinf100 abiuserp 0 2017-01-19 03:30 /ai/archive/prod/hdfs/inf/inf\_aj\_pvt/daily\_2017-01-19\_032641/registered\_source/registered\_source.aj.v\_contractors\_arriba

drwxr-xr-x - uzinf100 abiuserp 0 2017-01-19 03:30 /ai/archive/prod/hdfs/inf/inf\_aj\_pvt/daily\_2017-01-19\_032641/registered\_source/registered\_source.aj.v\_contractors\_arriba/.WORK-serial

-rw-r--r-- 3 uzinf100 abiuserp 411189 2017-01-19 03:30 /ai/archive/prod/hdfs/inf/inf\_aj\_pvt/daily\_2017-01-19\_032641/registered\_source/registered\_source.aj.v\_contractors\_arriba/000000.dat.gz

drwxr-xr-x - uzinf100 abiuserp 0 2017-01-19 03:30 /ai/archive/prod/hdfs/inf/inf\_aj\_pvt/daily\_2017-01-19\_032641/reject

-rw-r--r-- 3 uzinf100 abiuserp 20 2017-01-19 03:30 /ai/archive/prod/hdfs/inf/inf\_aj\_pvt/daily\_2017-01-19\_032641/reject/ingest.aj.v\_contractors\_arriba.dups.tech.dat.gz

drwxr-xr-x - uzinf100 abiuserp 0 2017-01-19 03:30 /ai/archive/prod/hdfs/inf/inf\_aj\_pvt/daily\_2017-01-19\_032641/today

drwxr-xr-x - uzinf100 abiuserp 0 2017-01-19 03:30 /ai/archive/prod/hdfs/inf/inf\_aj\_pvt/daily\_2017-01-19\_032641/today/today.aj.v\_contractors\_arriba

-rw-r--r-- 3 uzinf100 abiuserp 537016 2017-01-19 03:30 /ai/archive/prod/hdfs/inf/inf\_aj\_pvt/daily\_2017-01-19\_032641/today/today.aj.v\_contractors\_arriba/today.aj.v\_contractors\_arriba.000000.snappy.avro

pedgenode02:uzinf100:/ai/dropbox/prod/inf/inf\_aj\_pvt/incoming

**Exit Code**: Returns 0 on success and -1 on error.

## lsr

**Usage**: hdfs dfs -lsr <args>

Recursive version of ls. Similar to Unix ls -R.

## cat

**Usage**: hdfs dfs -cat URI [URI ...]

It is used to view contains of a file

**Example**:

hdfs dfs -cat hdfs://nn1.example.com/file1

hdfs dfs -cat file:///file3 /user/hadoop/file4

**Exit Code**: Returns 0 on success and -1 on error.

## count

**Usage**: hdfs dfs -count [-q] <paths>

Count the number of directories, files and bytes under the paths that match the specified file pattern. The output columns with -count are: DIR\_COUNT, FILE\_COUNT, CONTENT\_SIZE, FILE\_NAME

The output columns with -count -q are: QUOTA, REMAINING\_QUATA, SPACE\_QUOTA, REMAINING\_SPACE\_QUOTA, DIR\_COUNT, FILE\_COUNT, CONTENT\_SIZE, FILE\_NAME

**Example**:

> hadoop fs -count /ai/data/prod/hdfs/inf/inf\_aj\_pvt/main/today.aj

2 1 536339 /ai/data/prod/hdfs/inf/inf\_aj\_pvt/main/today.aj

pedgenode02:uzinf100:/ai/src/prod/global/sand/inf/inf\_new\_pvt

> hadoop fs -ls -R /ai/data/prod/hdfs/inf/inf\_aj\_pvt/main/today.aj

drwxrwxr-x - uzinf100 abiuserp 0 2017-01-30 17:01 /ai/data/prod/hdfs/inf/inf\_aj\_pvt/main/today.aj/today.aj.v\_contractors\_arriba

-rw-rw-r-- 3 uzinf100 abiuserp 536339 2017-01-30 17:01 /ai/data/prod/hdfs/inf/inf\_aj\_pvt/main/today.aj/today.aj.v\_contractors\_arriba/today.aj.v\_contractors\_arriba.000000.snappy.avro

pedgenode02:uzinf100:/ai/src/prod/global/sand/inf/inf\_new\_pvt

**Exit Code**: Returns 0 on success and -1 on error.

## chmod

**Usage**: hdfs dfs -chmod [-R] <MODE[,MODE]... | OCTALMODE> URI [URI ...]

Change the permissions of files. The user must be the owner of the file, or else a super-user.

Example:

hdfs -dfs -chmod 777 hdfs://nn1.example.com/file1

**Options**: The -R option will make the change recursively through the directory structure.

## getmerge

**Usage**: hdfs dfs -getmerge <src> <localdst> [addnl]

Takes a source directory and a destination file as input and concatenates files in src into the destination local file. Optionally addnl can be set to enable adding a newline character at the end of each file.

It is used for copying to localhost any hdfs file that containing multiple parts files.

**Example**:

hdfs dfs -getmerge /user/hadoop/sf-salaries-2011-2013/ /user/hadoop/sf-salaries-2014/ /root/output.csv

It merges the files in sf-salaries-2011-2013 and sf-salaries-2014 to output.csv in the root directory of the local filesystem.

## touchz

**Usage**: hdfs dfs -touchz URI [URI ...]

Create a file of zero length.

**Example**:

hadoop -touchz filename

**Exit Code**: Returns 0 on success and -1 on error.

## copyToLocal

**Usage**: hdfs dfs -copyToLocal [-ignorecrc] [-crc] URI <localdst>

Similar to get command, except that the destination is restricted to a local file reference.

**Example**:

hdfs dfs -copyToLocal /user/hadoop/file localfile

hdfs dfs -copyToLocal hdfs://nn.example.com/user/hadoop/file localfile

## get

**Usage**: hdfs dfs -get [-ignorecrc] [-crc] <src> <localdst>

Copy files to the local file system. Files that fail the CRC check may be copied with the -ignorecrc option. Files and CRCs may be copied using the -crc option.

**Example**:

hdfs dfs -get /user/hadoop/file localfile

hdfs dfs -get hdfs://nn.example.com/user/hadoop/file localfile

**Exit Code**: Returns 0 on success and -1 on error.

## appendToFile

**Usage**: hdfs dfs -appendToFile <localsrc> ... <dst>

Append single src, or multiple srcs from local file system to the destination file system. Also reads input from stdin and appends to destination file system.

**Example**:

hdfs dfs -appendToFile localfile /user/hadoop/hadoopfile

hdfs dfs -appendToFile localfile1 localfile2 /user/hadoop/hadoopfile

hdfs dfs -appendToFile localfile hdfs://nn.example.com/hadoop/hadoopfile

hdfs dfs -appendToFile - hdfs://nn.example.com/hadoop/hadoopfile Reads the input from stdin.

**Exit Code**: Returns 0 on success and 1 on error.

## copyFromLocal

**Usage**: hdfs dfs -copyFromLocal <localsrc> URI

Similar to put command, except that the source is restricted to a local file reference.

**Options**:

The -f option will overwrite the destination if it already exists.

**Example**:

hdfs dfs -copyFromLocal localfile /user/hadoop/hadoopfile

hdfs dfs -copyFromLocal localfile1 localfile2 /user/hadoop/hadoopdir

## put

**Usage**: hdfs dfs -put <localsrc> ... <dst>

Copy single src, or multiple srcs from local file system to the destination file system. Also reads input from stdin and writes to destination file system.

**Example**:

hdfs dfs -put localfile /user/hadoop/hadoopfile

hdfs dfs -put localfile1 localfile2 /user/hadoop/hadoopdir

hdfs dfs -put localfile hdfs://nn.example.com/hadoop/hadoopfile

hdfs dfs -put - hdfs://nn.example.com/hadoop/hadoopfile Reads the input from stdin.

**Exit Code**: Returns 0 on success and -1 on error.

## cp

**Usage**: hdfs dfs -cp [-f] URI [URI ...] <dest>

Copy files from source to destination. This command allows multiple sources as well in which case the destination must be a directory.

**Options**: The -f option will overwrite the destination if it already exists.

**Example**:

hdfs dfs -cp /user/hadoop/file1 /user/hadoop/file2

hdfs dfs -cp /user/hadoop/file1 /user/hadoop/file2 /user/hadoop/dir

**Exit Code**: Returns 0 on success and -1 on error.

## mv

**Usage**: hdfs dfs -mv URI [URI ...] <dest>

Moves files from source to destination. This command allows multiple sources as well in which case the destination needs to be a directory. Moving files across file systems is not permitted.

**Example**:

hdfs dfs -mv /user/hadoop/file1 /user/hadoop/file2

hdfs dfs -mv hdfs://nn.example.com/file1 hdfs://nn.example.com/file2 hdfs://nn.example.com/file3 hdfs://nn.example.com/dir1

**Exit Code**: Returns 0 on success and -1 on error.

## du

**Usage**: hdfs dfs -du [-s] [-h] URI [URI ...]

Displays sizes of files and directories contained in the given directory or the length of a file in case its just a file.

**Options**:

The -s option will result in an aggregate summary of file lengths being displayed, rather than the individual files.

The -h option will format file sizes in a "human-readable" fashion (e.g 64.0m instead of 67108864)

**Example**:

hdfs dfs -du /user/hadoop/dir1 /user/hadoop/file1 hdfs://nn.example.com/user/hadoop/dir1

**Exit Code**: Returns 0 on success and -1 on error.

## dus

**Usage**: hdfs dfs -dus <args>

It displays a summary of file lengths. This is an alternate form of hdfs dfs -du -s.

## mkdir

Usage: hdfs dfs -mkdir [-p] <paths>

Takes path uri's as argument and creates directories.

**Options**:

The -p option behavior is much like Unix mkdir -p, creating parent directories along the path.

**Example**:

hdfs dfs -mkdir /user/hadoop/dir1 /user/hadoop/dir2

hdfs dfs -mkdir hdfs://nn1.example.com/user/hadoop/dir hdfs://nn2.example.com/user/hadoop/dir

**Exit Code**: Returns 0 on success and -1 on error.

## rm

**Usage**: hdfs dfs -rm [-skipTrash] URI [URI ...]

Delete files specified as args. Only deletes non empty directory and files. If the -skipTrash option is specified, the trash, if enabled, will be bypassed and the specified file(s) deleted immediately. This can be useful when it is necessary to delete files from an over-quota directory. Refer to rmr for recursive deletes.

**Example**:

hdfs dfs -rm hdfs://nn.example.com/file /user/hadoop/emptydir

**Exit Code**: Returns 0 on success and -1 on error.

## stat

**Usage**: hdfs dfs -stat URI [URI ...]

Returns the stat information on the path.

**Example**:

hdfs dfs -stat path

**Exit Code**: Returns 0 on success and -1 on error.

## tail

**Usage**: hdfs dfs -tail [-f] URI

Displays last kilobyte of the file to stdout.

**Options**: The -f option will output appended data as the file grows, as in Unix.

**Example**:

hdfs dfs -tail pathname

**Exit Code**: Returns 0 on success and -1 on error.

## test

**Usage**: hdfs dfs -test -[ezd] URI

**Options**:

The -e option will check to see if the file exists, returning 0 if true.

The -z option will check to see if the file is zero length, returning 0 if true.

The -d option will check to see if the path is directory, returning 0 if true.

**Example**:

hdfs dfs -test -e filename