HDFS UNIX Commands

appendToFile

Usage: hdfs dfs -appendToFile …

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.

- 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.

cat

Usage: Hdfs dfs -cat URI [URI …]

Copies source paths to stdout.

Example:

- hdfs dfs -cat hdfs://nn1.example.com/file1 hdfs://nn2.example.com/file2
- hdfs dfs -cat file:///file3 /user/hadoop/file4

Exit Code:

Returns 0 on success and -1 on error.

checksum

Usage: hdfs dfs -checksum URI

Returns the checksum information of a file.

Example:

• hdfs dfs -checksum hdfs://nn1.example.com/file1

• hdfs dfs -checksum file:///etc/hosts

chgrp

Usage: hdfs dfs -chgrp [-R] GROUP URI [URI …]

Change group association of files. The user must be the owner of files, or else a super-user. Additional information is in the Permissions Guide.

Options

• The -R option will make the change recursively through the directory structure.

chmod

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

Change the permissions of files. With -R, make the change recursively through the directory structure. The user must be the owner of the file, or else a super-user. Additional information is in the Permissions Guide.

Options

• The -R option will make the change recursively through the directory structure.

chown

Usage: hdfs dfs -chown [-R] [OWNER][:[GROUP]] URI [URI]

Change the owner of files. The user must be a superuser. Additional information is in the Permissions Guide.

Options

• The -R option will make the change recursively through the directory structure.

copyFromLocal

Usage: hdfs dfs -copyFromLocal 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.

copyToLocal

Usage: hdfs dfs -copyToLocal [-ignorecrc] [-crc] URI

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

count

Usage: hdfs dfs -count [-q] [-h] [-v] 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, PATHNAME

The output columns with -count -q are: QUOTA, REMAINING_QUATA, SPACE_QUOTA, REMAINING_SPACE_QUOTA, DIR_COUNT, FILE_COUNT, CONTENT_SIZE, PATHNAME

The -h option shows sizes in human readable format.

The -v option displays a header line.

Example:

• hdfs dfs -count hdfs://nn1.example.com/file1 hdfs://nn2.example.com/file2

• hdfs dfs -count -q hdfs://nn1.example.com/file1

• hdfs dfs -count -q -h hdfs://nn1.example.com/file1

• hdfs dfs -count -q -h -v hdfs://nn1.example.com/file1

Exit Code:

Returns 0 on success and -1 on error.

cp

Usage: hdfs dfs -cp [-f] [-p | -p[topax]] URI [URI …]

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

'raw.*' namespace extended attributes are preserved if (1) the source and destination filesystems support them (HDFS only), and (2) all source and destination pathnames are in the /.reserved/raw hierarchy. Determination of whether raw.* namespace xattrs are preserved is independent of the -p (preserve) flag.

Options:

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

• The -p option will preserve file attributes [topx] (timestamps, ownership, permission, ACL, XAttr). If -p is specified with no arg, then preserves timestamps, ownership, permission. If -pa is specified, then preserves permission also because ACL is a super-set of permission. Determination of whether raw namespace extended attributes are preserved is independent of the -p flag.

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.

createSnapshot

See HDFS Snapshots Guide.

deleteSnapshot

See HDFS Snapshots Guide.

df

Usage: hdfs dfs -df [-h] URI [URI …]

Displays free space.

Options:

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

Example:

• hadoop dfs -df /user/hadoop/dir1

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 it’s 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

Displays a summary of file lengths.

Note: This command is deprecated. Instead, use hdfs dfs -du -s.

expunge

Usage: hdfs dfs -expunge

Empty the Trash. Refer to the HDFS Architecture Guide for more information on the Trash feature.

find

Usage: hdfs dfs -find … …

Finds all files that match the specified expression and applies selected actions to them. If no path is specified, then defaults to the current working directory. If no expression is specified, then defaults to -print.

The following primary expressions are recognized:

• -name pattern

-iname pattern

Evaluates as true if the basename of the file matches the pattern using standard file system globbing. If -iname is used then the match is case insensitive.

• -print

-print0Always

evaluates to true. Causes the current pathname to be written to standard output. If the -print0 expression is used, then an ASCII NULL character is appended.

The following operators are recognized:

• expression -a expression

expression -and expression

expression expression

Logical AND operator for joining two expressions. Returns true if both child expressions return true. Implied by the juxtaposition of two expressions and so does not need to be explicitly specified. The second expression will not be applied if the first fails.

Example:

hdfs dfs -find / -name test -print

Exit Code:

Returns 0 on success and -1 on error.

get

Usage: hdfs dfs -get [-ignorecrc] [-crc]

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.

getfacl

Usage: hdfs dfs -getfacl [-R] Displays the Access Control Lists (ACLs) of files and directories. If a directory has a default ACL, then getfacl also displays the default ACL.

Options:

• -R: List the ACLs of all files and directories recursively.

• path: File or directory to list.

Examples:

• hdfs dfs -getfacl /file

• hdfs dfs -getfacl -R /dir

Exit Code:

Returns 0 on success and non-zero on error.

getfattr

Usage: hdfs dfs -getfattr [-R] -n name | -d [-e en] Displays the extended attribute names and values (if any) for a file or directory.

Options:

• -R: Recursively list the attributes for all files and directories.

• -n name: Dump the named extended attribute value.

• -d: Dump all extended attribute values associated with the pathname.

• -e encoding: Encode values after retrieving them. Valid encodings are “text”, “hex”, and “base64”. Values encoded as text strings are enclosed in double quotes (“), and values encoded as hexadecimal and base64 are prefixed with 0x and 0s, respectively.

• path: The file or directory.

Examples:

• hdfs dfs -getfattr -d /file

• hdfs dfs -getfattr -R -n user.myAttr /dir

Exit Code:

Returns 0 on success and non-zero on error.

getmerge

Usage: hdfs dfs -getmerge [-nl]

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

Examples:

• hdfs dfs -getmerge -nl /src /opt/output.txt

• hdfs dfs -getmerge -nl /src/file1.txt /src/file2.txt /output.txt

Exit Code:

Returns 0 on success and non-zero on error.

help

Usage: hdfs dfs -help

Return usage output.

ls

Usage: hdfs dfs -ls [-d] [-h] [-R]

Options:

• -d: Directories are listed as plain files.

• -h: Format file sizes in a human-readable fashion (e.g. 64.0m instead of 67108864).

• -R: Recursively list subdirectories encountered.

For a file ls 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

Files within a directory are order by filename by default.

Example:

• hdfs dfs -ls /user/hadoop/file1

Exit Code:

Returns 0 on success and -1 on error.

lsr

Usage: hdfs dfs -lsr

Recursive version of ls.

Note: This command is deprecated. Instead, use hdfs dfs -ls -R

mkdir

Usage: hdfs dfs -mkdir [-p] Takes path uri's as an 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.

moveFromLocal

Usage: hdfs dfs -moveFromLocal

Similar to put command, except that the source localsrc is deleted after it's copied.

moveToLocal

Usage: hdfs dfs -moveToLocal [-crc]

Displays a "Not implemented yet" message.

mv

Usage: hdfs dfs -mv URI [URI …]

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.

put

Usage: hdfs dfs -put …

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.

• 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.

renameSnapshot

See HDFS Snapshots Guide.

rm

Usage: hdfs dfs -rm [-f] [-r |-R] [-skipTrash] URI [URI …]

Delete files specified as args.

Options:

• The -f option will not display a diagnostic message or modify the exit status to reflect an error if the file does not exist.

• The -R option deletes the directory and any content under it recursively.

• The -r option is equivalent to -R.

• The -skipTrash option will bypass trash if enabled, and delete the specified file(s) immediately. This can be useful when it is necessary to delete files from an over-quota directory.

Example:

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

Exit Code:

Returns 0 on success and -1 on error.

rmdir

Usage: hdfs dfs -rmdir [–ignore-fail-on-non-empty] URI [URI …]

Delete a directory.

Options:

• –ignore-fail-on-non-empty: When using wildcards, do not fail if a directory still contains files.

Example:

• hdfs dfs -rmdir /user/hadoop/emptydir

rmr

Usage: hdfs dfs -rmr [-skipTrash] URI [URI …]

Recursive version of delete.

Note: This command is deprecated. Instead use hdfs dfs -rm -r

setfacl

Usage: hdfs dfs -setfacl [-R] [-b |-k -m |-x] |[–set]

Sets Access Control Lists (ACLs) of files and directories.

Options:

• -b: Remove all but the base ACL entries. The entries for user, group and others are retained for compatibility with permission bits.

• -k: Remove the default ACL.

• -R: Apply operations to all files and directories recursively.

• -m: Modify ACL. New entries are added to the ACL, and existing entries are retained.

• -x: Remove specified ACL entries. Other ACL entries are retained.

• –set: Fully replaces the ACL, discarding all existing entries. The acl_spec must include entries for user, group, and others for compatibility with permission bits.

• acl_spec: Comma separated list of ACL entries.

• path: File or directory to modify.

Examples:

• hdfs dfs -setfacl -m user: hadoop:rw- /file

• hdfs dfs -setfacl -x user: hadoop /file

• hdfs dfs -setfacl -b /file

• hdfs dfs -setfacl -k /dir

• hdfs dfs -setfacl –set user::rw-,user:hadoop:rw-,group::r–,other::r– /file

• hdfs dfs -setfacl -R -m user:hadoop:r-x /dir

• hdfs dfs -setfacl -m default:user:hadoop:r-x /dir

Exit Code:

Returns 0 on success and non-zero on error.

setfattr

Usage: hdfs dfs -setfattr -n name [-v value] | -x name Sets an extended attribute name and value for a file or directory.

Options:

• -b: Remove all but the base ACL entries. The entries for user, group and others are retained for compatibility with permission bits.

• -n name: The extended attribute name.

• -v value: The extended attribute value. There are three different encoding methods for the value. If the argument is enclosed in double quotes, then the value is the string inside the quotes. If the argument is prefixed with 0x or 0X, then it is taken as a hexadecimal number. If the argument begins with 0s or 0S, then it is taken as a base64 encoding.

• -x name: Remove the extended attribute.

• path: The file or directory.

Examples:

• hdfs dfs -setfattr -n user.myAttr -v myValue /file

• hdfs dfs -setfattr -n user.noValue /file

• hdfs dfs -setfattr -x user.myAttr /file

Exit Code:

Returns 0 on success and non-zero on error.

setrep

Usage: hdfs dfs -setrep [-R] [-w] Changes the replication factor of a file. If path is a directory, then the command recursively changes the replication factor of all files under the directory tree rooted at path.

Options:

• The -w flag requests that the command waits for the replication to complete. This can potentially take a very long time.

• The -R flag is accepted for backward compatibility. It has no effect.

Example:

• hdfs dfs -setrep -w 3 /user/hadoop/dir1

Exit Code:

Returns 0 on success and -1 on error.

stat

Usage: hdfs dfs -stat [format] …

Print statistics about the file/directory at in the specified format. Format accepts file size in blocks (%b), type (%F), the group name of owner (%g), name (%n), block size (%o), replication (%r), the user name of owner(%u), and modification date (%y, %Y). %y shows UTC date as “yyyy-MM-dd HH:mm:ss” and %Y shows milliseconds since January 1, 1970, UTC. If the format is not specified, %y is used by default.

Example:

• hdfs dfs -stat “%F %u:%g %b %y %n” /file

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 -[defsz] URI

Options:

• -d: f the path is a directory, return 0.

• -e: if the path exists, return 0.

• -f: if the path is a file, return 0.

• -s: if the path is not empty, return 0.

• -z: if the file is zero length, return 0.

Example:

• hdfs dfs -test -e filename

text

Usage: hdfs dfs -text

Takes a source file and outputs the file in text format. The allowed formats are zip and TextRecordInputStream.

touchz

Usage: hdfs dfs -touchz URI [URI …]

Create a file of zero length.

Example:

• hdfs dfs -touchz pathname

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

truncate

Usage: hdfs dfs -truncate [-w] Truncate all files that match the specified file pattern to the specified length.

Options:

• The -w flag requests that the command waits for block recovery to complete, if necessary. Without -w flag the file may remain unclosed for some time while the recovery is in progress. During this time file cannot be reopened for append.

Example:

• hdfs dfs -truncate 55 /user/hadoop/file1 /user/hadoop/file2

• hdfs dfs -truncate -w 127 hdfs://nn1.example.com/user/hadoop/file1

usage

Usage: hdfs dfs -usage command

Return the help for an individual command.