Overview

The File System (FS) shell includes various shell-like commands that directly interact with the Hadoop Distributed File System (HDFS) as well as other file systems that Hadoop supports, such as Local FS, HFTP FS, S3 FS, and others. The FS shell is invoked by:

bin/hadoop fs <args>

All FS shell commands take path URIs as arguments. The URI format is scheme://authority/path. For HDFS the scheme is hdfs, and for the Local FS the scheme is file. The scheme and authority are optional. If not specified, the default scheme specified in the configuration is used. An HDFS file or directory such as /parent/child can be specified as hdfs://namenodehost/parent/child or simply as /parent/child (given that your configuration is set to point to hdfs://namenodehost).

Most of the commands in FS shell behave like corresponding Unix commands. Differences are described with each of the commands. Error information is sent to stderr and the output is sent to stdout.

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.

- 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.

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 super-user. 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 <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.

copyToLocal

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

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

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:

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

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.

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>

Displays a summary of file lengths. This is an alternate form of 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.

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.

getfacl

Usage: hdfs dfs -getfacl [-R] <path>

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.

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.

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

Example:

- hdfs dfs -ls /user/hadoop/file1

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.

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.

moveFromLocal

Usage: dfs -moveFromLocal <localsrc> <dst>

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

moveToLocal

Usage: hdfs dfs -moveToLocal [-crc] <src> <dst>

Displays a "Not implemented yet" message.

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.

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.

- 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.

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.

rmr

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

Recursive version of delete. 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.

Example:

- hdfs dfs -rmr /user/hadoop/dir
- hdfs dfs -rmr hdfs://nn.example.com/user/hadoop/dir

Exit Code:

Returns 0 on success and -1 on error.

setfacl

Usage: hdfs dfs -setfacl [-R] [-b|-k -m|-x <acl_spec> <path>]|[--set <acl_spec> <path>]

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 replace 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.

setrep

Usage: hdfs dfs -setrep [-R] [-w] <numReplicas> <path>

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 wait for the replication to complete. This can potentially take a very long time.
- The -R flag is accepted for backwards 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 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

text

Usage: hdfs dfs -text <src>

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:

- hadoop -touchz pathname

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