hadoop fs

The `hadoop fs` command runs a generic filesystem user client that interacts with the MapR filesystem (MapR-FS).

On the Windows client, make sure that the `PATH` contains the following directories:

- `C:\Windows\system32`
- `C:\Windows`

If they are not present, the hadoop fs command might fail silently.

Syntax

```
hadoop [ Generic Options ] fs
        [-cat <src>]
 [-chgrp [-R] GROUP PATH...]
 [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
 [-chown [-R] [OWNER][:[GROUP]] PATH...]
        [-copyFromLocal <localsrc> ... <dst>]
        [-copyToLocal [-ignoreCrc] [-crc] <src> <localdst>]
        [-count[-q] <path>]
        [-cp <src> <dst>]
        [-df <path>]
        [-du <path>]
 [-dus <path>]
        [-expunge]
        [-get [-ignoreCrc] [-crc] <src> <localdst>
 [-getmerge <src> <localdst> [addnl]]
        [-help [cmd]]
        [-ls <path>]
        [-lsr <path>]
        [-mkdir <path>]
        [-moveFromLocal <localsrc> ... <dst>]
        [-moveToLocal <src> <localdst>]
        [-mv <src> <dst>]
        [-put <localsrc> ... <dst>]
        [-rm [-skipTrash] <src>]
 [-rmr [-skipTrash] <src>]
        [-stat [format] <path>]
 [-tail [-f] <path>]
        [-test -[ezd] <path>]
        [-text <path>]
 [-touchz <path>]
```

Parameters

Command Options

The following command parameters are supported for `hadoop fs`:

Parameter	Description
`-cat <src>`	Fetch all files that match the file pattern defined by the `<src>` parameter and display their contents on *stdout*.

`-fs [local | <file system URI>]`	Specify the file system to use. If not specified, the current configuration is used, taken from the following, in increasing precedence: `core-default.xml` inside the hadoop jar file `core-site.xml` in `$HADOOP_CONF_DIR` The `local` option means use the local file system as your DFS. `<file system URI>` specifies a particular file system to contact. This argument is optional but if used must appear appear first on the command line. Exactly one additional argument must be specified.
`-ls <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 `dirName (full path) <dir>` and file entries are of the form `fileName(full path) <r n>`. size where n is the number of replicas specified for the file and size is the size of the file, in bytes.
`-lsr <path>`	Recursively list the contents that match the specified file pattern. Behaves very similarly to `hadoop fs -ls`, except that the data is shown for all the entries in the subtree.
`-df [<path>]`	Shows the capacity, free and used space of the filesystem. If the filesystem has multiple partitions, and no path to a particular partition is specified, then the status of the root partitions will be shown.
`-du <path>`	Show the amount of space, in bytes, used by the files that match the specified file pattern. Equivalent to the Unix command `du -sb <path>/*` in case of a directory, and to `du -b <path>` in case of a file. The output is in the form name (full path) size (in bytes).
`-dus <path>`	Show the amount of space, in bytes, used by the files that match the specified file pattern. Equivalent to the Unix command `du -sb`. The output is in the form `name(full path)` size (in bytes).
`-mv <src> <dst>`	Move files that match the specified file pattern `<src>` to a destination `<dst>`. When moving multiple files, the destination must be a directory.
`-cp <src> <dst>`	Copy files that match the file pattern `<src>` to a destination. When copying multiple files, the destination must be a directory.
`-rm [-skipTrash] <src>`	Delete all files that match the specified file pattern. Equivalent to the Unix command `rm <src>`. If enabled, the `-skipTrash` option bypasses trashand immediately deletes `<src>`.
`-rmr [-skipTrash] <src>`	Remove all directories that match the specified file pattern. Equivalent to the Unix command `rm -rf <src>`. If enabled, the `-skipTrash` option bypasses trashand immediately deletes `<src>`.
`-put <localsrc> ... <dst>`	Copy files from the local file system into fs.
`-copyFromLocal <localsrc> ... <dst>`	Identical to the `-put` command.
`-moveFromLocal <localsrc> ... <dst>`	Same as `-put`, except that the source is deleted after it's copied.
`-get [-ignoreCrc] [-crc] <src> <localdst>`	Copy files that match the file pattern <src> to the local name. <src> is kept. When copying multiple files, the destination must be a directory.

`-getmerge <src> <localdst>`	Get all the files in the directories that match the source file pattern and merge and sort them to only one file on local fs. `<src>` is kept.
`-copyToLocal [-ignoreCrc] [-crc] <src> <localdst>`	Identical to the `-get` command.
`-moveToLocal <src> <localdst>`	Not implemented yet.
`-mkdir <path>`	Create a directory in specified location.
`-tail [-f] <file>`	Show the last 1KB of the file. The `-f` option shows appended data as the file grows.
`-touchz <path>`	Write a timestamp in `yyyy-MM-dd HH:mm:ss` format in a file at `<path>`. An error is returned if the file exists with non-zero length.
`-test -[ezd] <path>`	If file { exists, has zero length, is a directory then return 0, else return 1.
`-text <src>`	Takes a source file and outputs the file in text format. The allowed formats are `zip` and TextRecordInputStream.
`-stat [format] <path>`	Print statistics about the file/directory at `<path>` in the specified format. Format accepts filesize in blocks (%b), filename (%n), block size (%o), replication (%r), and modification date (%y, %Y).
`-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...`	Changes permissions of a file. This works similar to shell's `chmod` with a few exceptions. `-R` modifies the files recursively. This is the only option currently supported. `MODE` is same as the mode used for `chmod` shell command. The only letters recognized are `rwxXt`. That is, `+t,a+r,g-w,+rwx,o=r` `OCTALMODE` is specifed in 3 or 4 digits. If 4 digits, the first may be 1 or 0 to turn the sticky bit on or off, respectively. Unlike shell command, it is not possible to specify only part of the mode For example, `754` is same as `u=rwx,g=rx,o=r` If none of 'augo' is specified, 'a' is assumed and unlike shell command, no umask is applied.
`-chown [-R] [OWNER][:[GROUP]] PATH...`	Changes owner and group of a file. This is similar to shell's `chown` with a few exceptions. `-R` modifies the files recursively. This is the only option currently supported. If only owner or group is specified then only owner or group is modified. The owner and group names may only consists of digits, alphabet, and any of `-.@/` *i.e.* `[-.@/a-zA-Z0-9]`. The names are case-sensitive. **Warning** WARNING: Avoid using '.' to separate user name and group though Linux allows it. If user names have dots in them and you are using local file system, you might see surprising results sinceshell command `chown` n is used for local files.
`-chgrp [-R] GROUP PATH...`	This is equivalent to `-chown ... :GROUP ...`
`-count[-q] <path>`	Count the number of directories, files and bytes under the paths that match the specified file pattern. The output columns are: • `DIR_COUNT FILE_COUNT CONTENT_SIZE FILE_NAME` or • `QUOTA REMAINING_QUATA SPACE_QUOTA REMAINING_SPACE_QUOTA` • `DIR_COUNT FILE_COUNT CONTENT_SIZE FILE_NAME`

`-help [cmd]`	Displays help for given command or all commands if none is specified.

Generic Options

The following generic options are supported for the `hadoop fs` command: `-conf <configuration file>`, `-D <property=value>`, `-fs <local|file system URI>`, `-jt <local|jobtracker:port>`, `-files <file1,file2,file3,...>`, `-libjars <libjar1,libjar2,libjar3,...>`, and `-archives <archive1,archive2,archive3,...>`.

For more information on generic options, see Generic Options.