

hadoop mfs

The *hadoop mfs* command performs operations on files and directories in the cluster.

The main purposes of `hadoop mfs` are to display directory information and contents, to create symbolic links and hard links, to set, get, and remove Access Control Expressions (ACE) on files and directories, and to set compression and chunk size on a directory.

Syntax

hadoop mfs

```
[ -count <path> ]
[ -delace [-R] <path> ]
[ -getace [-R] <path> ]
[ -help <command> ]
[ -ln <target> <symlink> ]
[ -lnh <target> <hardlink> ]
[ -ls <path> ]
[ -lsd <path> ]
[ -lso <path> ]
[ -lsor <path> ]
[ -lsr <path> ]
[ -Lsr <path> ]
[ -lsrv <path> ]
[ -lss <path> ]
[ -offload <file_path> [-v] ]
[ -recall <file_path> [-v] ]
[ -rmr <path> ]
[ -setace [-R]
  [-readfile <ace>] [-writefile <ace>] [-executefile <ace>]
  [-addchild <ace>] [-deletechild <ace>] [-lookupdir <ace>] [-readdir <ace>]
  [-aces "[rf:<ace>],[wf:<ace>],[ef:<ace>],[ac:<ace>],[dc:<ace>],[rd:<ace>],[ld:<ace>]" ]
  [-preservemodebits <true|false>] [-setinherit <true|false>] <path> ]
[ -setaudit on|off <dir|file|table> ]
[ -setcompression on|off|lzf|lz4|zlib <dir|table> ]
[ -setchunksize <size> <dir> ]

[ -setnetworkencryption on|off <target> ]
[ -stat <path> ]
[ -tierstatus <file_path> [-v] ]
```

Copy

Parameters

The normal command syntax is to specify a single option from the following table, along with its corresponding arguments. If compression and chunk size are not set explicitly for a given directory, the values are inherited from the parent directory.

Parameter	Description
-count <path>	Counts and returns the number of directories and (regular, symbolic link, volume link, kvstores, and device) files in the specified path (recursively).
-delace [-R] <path>	<p>Deletes all ACEs (../SecurityGuide/ACEs.html) associated with the specified file or directory and sets ACEs for the specified file or directory to the default value, which is the empty string. Here:</p> <ul style="list-style-type: none">[-R] — Enables recursion allowing you to perform the operation in subdirectories as well.<path> — Specifies path to file or directory. <p>Note: You cannot delete specific access types with this parameter. Instead, if necessary, reset the value for the specific access type to the empty string using <code>-setace</code> parameter. If the empty string is used to deny a specific type of access, that type of access will be denied to all users. To deny specific types of access to specific users only, use the negation operator (!). The mode bits corresponding to the ACEs being deleted will not change.</p>
-getace [-R] <path>	<p>Returns the permissions -- POSIX mode bits and ACEs (../SecurityGuide/ACEs.html) -- for the given file or (recursively) for the directory. Recursion is enabled only if <code>-R</code> is specified; if <code>-R</code> is not specified, returns the permissions for the given directory only. Here:</p> <ul style="list-style-type: none">[-R] — (Optional) Enables recursion allowing you to perform the operation in subdirectories as well.<path> — (Required) Specifies path to file or directory. <p>Note: If one or more ACEs are available for the file or directory, a plus sign (+), which indicates that both ACEs and POSIX mode bits are set for the given file or directory, will also be returned. If the ACE on the file or directory is an empty string, the plus sign will not be returned.</p>

Parameter	Description
-help <command>	Displays help for the <code>hadoop mfs</code> command.
-ln <target> <symlink>	Creates a symbolic link <symlink> that points to the target path <target> , similar to the standard Linux <code>ln -s</code> command.
-lnh <target> <hardlink>	Creates a hardlink that associates a new name/file path with an existing file. You must specify the following: <target> File name, including the full path, of the file to link to. <hardlink> New name, including the full path, of the file to link with.
-ls <path>	Lists files in the directory specified by <path> . The <code>hadoop mfs -ls</code> command corresponds to the standard <code>hadoop fs -ls</code> command, but provides the following additional information: <ul style="list-style-type: none"> • Chunks used for each file • Server where each chunk resides • Whether compression is enabled for each file • Whether encryption is enabled for each file • Whether audit is enabled (A) or disabled (U) for each file
-lsd <path>	Lists files in the directory specified by <path> , and also provides information about the specified directory itself: <ul style="list-style-type: none"> • Whether compression is enabled for the directory (indicated by <code>z</code>) • The configured chunk size (in bytes) for the directory.
-lso <path>	Lists files in the directory specified by <path> . The <code>hadoop mfs -lso</code> command corresponds to the standard <code>hadoop fs -ls</code> command, but provides the following additional information: <ul style="list-style-type: none"> • Whether compression is enabled for each file • Whether encryption is enabled for each file • Whether audit is enabled (A) or disabled (U) for each file This command is faster than <code>hadoop fs -ls</code> as it uses optimized printing method to dump data on screen.
-lsor <path>	Lists files in the directory specified by <path> recursively. This command is the recursive variant of the <code>hadoop mfs -lso</code> command.
-lsr <path>	Lists files in the directory and subdirectories specified by <path> , recursively, including dereferencing symbolic links. The <code>hadoop mfs -lsr</code> command corresponds to the standard <code>hadoop fs -lsr</code> command, but provides the following additional information: <ul style="list-style-type: none"> • Chunks used for each file • Server where each chunk resides
-Lsr <path>	Equivalent to <code>lsr</code> , but additionally dereferences symbolic links
-lsrv <path>	Lists all paths recursively without crossing volume links.
-lss <path>	Lists files in the directory specified by <path> , with an additional column that displays the number of disk blocks per file. Disk blocks are 8192 bytes.
offload <file_path> [-v]	The file to offload to the storage tier. This is a blocking operation; the control is not returned until the operation is complete and the file has been offloaded. Use <code>-v</code> (for verbose) to view the status of the ongoing offload.
recall <file_path> [-v]	The file to recall from the storage tier. This is a blocking operation; the control is not returned until the operation is complete and the file has been recalled. Use <code>-v</code> (for verbose) to view the status of the ongoing recall.
-rmr <path>	Deletes files and directories in the specified path recursively. This is a highly optimized version of the normal generic <code>hadoop fs rmr</code> command and is 10X faster for large directories. This option is useful when one or more directories in the specified path contains many (millions of) files.
-setace [-R] [-readfile <ace>] [-writefile <ace>] [-executefile <ace>] [-addchild <ace>] [-deletechild <ace>] [-lookupdir <ace>]	Sets or modifies the read, write, and/or execute permissions [../SecurityGuide/ACEs.html] for files or directories. This argument will <ul style="list-style-type: none"> • Overwrite existing values with new values, if specified, for access types that were previously set. • Set new values for access types that have not yet been set. • Not set or modify access types that were not passed in with the command, whether they were previously set or are unset.

Parameter	Description														
<pre>[-readdir <ace>] [-aces "[rf:<ace>],[wf:<ace>],[ef:<ace>],[ac:<ace>],[dc:<ace>],[rd:<ace>],[ld:<ace>]"] [-preservemodebits <true false>] [-setinherit <true false>] <path></pre>	<p>Note: ACEs must be specified immediately after the <code>-setace</code> parameter. All other arguments, if specified, must be given after the ACEs.</p> <p>Here:</p> <p>-R Enables recursion allowing you to perform the operation in subdirectories as well. Recursion is enabled only if <code>-R</code> is specified; if <code>-R</code> is not specified, sets or modifies the permissions for the given directory only.</p> <p><ace> Access Control Expressions (../SecurityGuide/SyntaxOfACE.html#SyntaxofAccessControlExpressions) defined using boolean expressions and sub-expressions.</p> <p>-readfile -writefile -executefile Specifies permissions (defined using ACE) for reading, writing, and/or executing the file.</p> <p>-readdir -lookupdir Specifies permissions (defined using ACE) for accessing the directory. To read the files in the directory, the user must be granted <code>lookupdir</code> access and to write to or execute files in the directory, the user must be granted <code>readdir</code> and <code>lookupdir</code> access types.</p> <p>-addchild -deletechild Specifies permissions (defined using ACE) for adding or deleting subdirectories.</p> <p>-aces Specifies ACEs as a single string. Specify a comma-separated list of ACEs within quotes. The following permissions can be set.</p> <table border="1"> <tr> <td><code>rf</code></td><td>Refers to read file access.</td></tr> <tr> <td><code>wf</code></td><td>Refers to write file access</td></tr> <tr> <td><code>ef</code></td><td>Refers to execute file access</td></tr> <tr> <td><code>rd</code></td><td>Refers to read directory access. Note: To write to or execute files in the directory, the user must be granted both <code>rd</code> and <code>ld</code> permissions.</td></tr> <tr> <td><code>ld</code></td><td>Refers to lookup directory access. Note: To read the files in the directory, the user must be granted <code>ld</code> access and to write to or execute files in the directory, the user must be granted both <code>rd</code> and <code>ld</code> permissions.</td></tr> <tr> <td><code>ac</code></td><td>Refers to add child directory access.</td></tr> <tr> <td><code>dc</code></td><td>Refers to delete child directory access.</td></tr> </table> <p>Note: ACEs can be up to 60KB in length.</p> <p>-preservemodebits Indicates whether to preserve or modify POSIX mode bits. Values can be:</p> <ul style="list-style-type: none"> <code>true</code> - preserve existing POSIX mode bits <code>false</code> - change POSIX mode bits to match the given ACE setting <p>By default, this is set to <code>false</code>.</p> <p>-setinherit Indicates whether or not to allow all files and subdirectories under the specified directory to inherit the ACE of the parent directory. Values can be:</p> <ul style="list-style-type: none"> <code>true</code> - inherit ACE from parent <code>false</code> - do not inherit ACE from parent <p>By default, this is set to <code>true</code>. If <code>true</code>, new files and directories under the parent will inherit the ACEs and corresponding POSIX mode bits from the parent directory.</p>	<code>rf</code>	Refers to read file access.	<code>wf</code>	Refers to write file access	<code>ef</code>	Refers to execute file access	<code>rd</code>	Refers to read directory access. Note: To write to or execute files in the directory, the user must be granted both <code>rd</code> and <code>ld</code> permissions.	<code>ld</code>	Refers to lookup directory access. Note: To read the files in the directory, the user must be granted <code>ld</code> access and to write to or execute files in the directory, the user must be granted both <code>rd</code> and <code>ld</code> permissions.	<code>ac</code>	Refers to add child directory access.	<code>dc</code>	Refers to delete child directory access.
<code>rf</code>	Refers to read file access.														
<code>wf</code>	Refers to write file access														
<code>ef</code>	Refers to execute file access														
<code>rd</code>	Refers to read directory access. Note: To write to or execute files in the directory, the user must be granted both <code>rd</code> and <code>ld</code> permissions.														
<code>ld</code>	Refers to lookup directory access. Note: To read the files in the directory, the user must be granted <code>ld</code> access and to write to or execute files in the directory, the user must be granted both <code>rd</code> and <code>ld</code> permissions.														
<code>ac</code>	Refers to add child directory access.														
<code>dc</code>	Refers to delete child directory access.														

Parameter	Description
	<p>If <code>false</code>, files and directories under the parent will not inherit the ACE of the parent directory. Instead, by default, the files and directories will get the default ACE, which is the empty string for all access types; POSIX mode bits will be set on the files and directories in the traditional way.</p> <p><path> Specifies path to file or directory.</p>
<code>-setaudit on off <dir file table></code>	<p>Enables auditing of the specified directory, file, or MapR Database table.</p> <p>Enabling auditing of a directory does not enable auditing of files and subdirectories that exist in the directory. You must enable auditing on those existing files and subdirectories. However, any new files and subdirectories that you create will automatically be enabled for auditing. See Checking Whether Auditing is Enabled for a Directory, File, or MapR Database Table (../SecurityGuide/CheckingWhetherAuditingisEnabled.html#CheckingWhetherAuditingisEnabledforaDirectoryFileorMapR-DBTable).</p> <p>For operations on the object to be logged, auditing also needs to be enabled on the cluster and the volume in which the object is located. See Enabling Auditing (../SecurityGuide/Auditing-Prerequisites.html#EnablingAuditing) for details. If auditing is enabled for a directory, new files and directories created within that directory are also enabled for auditing.</p>
<code>-setchunksize <size> <dir></code>	Sets the chunk size in bytes for the directory specified in <code><dir></code> . The <code><size></code> parameter must be a multiple of 65536.
<code>-setcompression on off lzf lz4 zlib <dir table></code>	<p>Turns compression on or off on the directory specified in <code><dir></code> or on the specified table, and sets the compression type to one of the following if compression is not turned off:</p> <ul style="list-style-type: none"> <code>on</code> — turns on compression using the default algorithm (LZ4) <code>lzf</code> — turns on compression and sets the algorithm to LZF <code>lz4</code> — turns on compression and sets the algorithm to LZ4 <code>zlib</code> — turns on compression and sets the algorithm to ZLIB
<code>-setnetworkencryption on off <target></code>	Sets network encryption on or off for the filesystem object defined in <code><target></code> . The cluster encrypts network target to or from a file, directory, stream, or MapR table with network security enabled.
<code>-stat <path></code>	<p>Displays the statistics for the given file. Only the root user and the MAPR_USER user (user name under which MapR services run) have permissions to run this command.</p> <p>The path is required and specifies the path (to the file) on which to run the command. The output fields for this command are described below (hadoop-mfs.html#hadoopmfs__stat-output-fields).</p>
<code>tierstatus <file_path> [-v]</code>	<p>The status of the offload or recall of the given file. If <code>-v</code> (for verbose) is also specified, for the given file, the command specifies whether data is local or offloaded as the final output. If the file:</p> <ul style="list-style-type: none"> Contains local data, returns the following final output: <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;">File has local data</div> <ul style="list-style-type: none"> Is completely offloaded, returns the following final output: <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;">File does not have local data</div> <p>See Output below for more information.</p>

Output

When used with `-ls`, `-lsd`, `-lso`, `-lsor`, `-lsr`, `-lss`, `hadoop mfs` displays information about files and directories. For each file or directory `hadoop mfs` displays a line of basic information followed by lines listing the chunks that make up the file, in the following format:

```
{mode} {compression} {encryption} {audit} {diskFlush} {replication} {owner} {group} {size} {date} {chunk size} {name} {chunk} {fid}
{host} [{host}...] {chunk} {fid} {host} [{host}...] ...
```

Volume links are displayed as follows:

```
{mode} {compression} {encryption} {audit} {diskFlush} {replication} {owner} {group} {size} {date} {chunk size} {name} {chunk} {target}
volume name} {writability} {fid} -> {fid} [{host}...]
```

The following table describes the values:

mode	A text string indicating the read, write, and execute permissions for the owner, group, and other permissions. See also Managing Permissions (../AdministratorGuide/c_setting_user_permissions_overview.html#SettingUserPermissions-Se_26982517-d3e207).
compression	<ul style="list-style-type: none"> U: uncompressed L: LZf Z (Uppercase): LZ4 z (Lowercase): ZLIB
encryption	U: unencrypted; E: encrypted
audit	U: disabled; A: enabled
disk flush	U:disabled; F:enabled
replication	The replication factor of the file (directories display a dash instead)
owner	The owner of the file or directory
group	The group of the file or directory
size	The size of the file or directory
date	The date the file or directory was last modified
chunk size	The chunk size of the file or directory
name	The name of the file or directory
chunk	The chunk number. The first chunk is a primary chunk labeled " p ", a 64K chunk containing the root of the file. Subsequent chunks are numbered in order.
fid	<p>The chunk's file ID, which consists of three parts:</p> <ul style="list-style-type: none"> The ID of the container where the file is stored The inode of the file within the container An internal version number <p>Note: For volume links, the first <code>fid</code> is the chunk that stores the volume link itself; the <code>fid</code> after the arrow (<code>-></code>) is the first chunk in the target volume.</p>
host	The host on which the chunk resides. When several hosts are listed, the first host is the first copy of the chunk and subsequent hosts are replicas.
target volume name	The name of the volume pointed to by a volume link.
writability	Displays whether the volume is writable.

When used with `-stat <path>`, `hadoop mfs` displays statistics for the given file. For each file, it displays the following:

Output field	Description
uid	The user ID of the owner.
atime	The last access time.
mtime	The last modified time.
nlink	The number of hard links.

Output field	Description
type	The type of the file. Value can be: <ul style="list-style-type: none">regulardirectorysymlinkvollinkkvstoredevice
size	The size of the file/directory. Depending on the type of file, it can be the actual size or the number of entries.
mode	The UNIX style permission mode bits for the file/directory.
networkencryption	The network encryption setting. If the network encryption is enabled for this file or not.
subtype	The subtype for the specified type. The following subtypes are supported for some of the types: regular <ul style="list-style-type: none">FSTRegBucketFSTRegCFFSTRegKeyMap kvstore <ul style="list-style-type: none">FSTKvTableFSTKvTabletMapFSTKvSchemaFSTKvTabletFSTKvSegMapFSTKvSpillMapFSTKvKeyMapFSTKvXattr device <ul style="list-style-type: none">FSTDevPipeFSTDevSocketFSTDevBlockFSTDevChar Note: For all other types, subtypes are not valid.
gid	The group ID.
compression	The compression setting.

When used with `tierstatus` , the output varies based on whether or not data is local, was offloaded, or was recalled. The output looks similar to the following if:

- Data was completely offloaded:

File does not have local data

- Data could not be completely offloaded or data was recalled:

File has local data

Examples

View File Information

The `hadoop mfs` command is used to view file contents. You can use this command to check if compression is turned off in a directory or mounted volume. For example,

```
# hadoop mfs -ls /
Found 121 items
vrwxr-xr-x  Z E U U   3 mapr mapr          121 2018-08-10 01:07 268435456 /.rw
p mapr.cluster.root writeable 2049.50.131362 -> 2049.16.2  physical19.qa.lab:5660 physical20.qa.lab:5660 physical
123.qa.lab:5660
vrwxr-xr-x  Z E U U   3 root root          1 2018-08-09 19:26 268435456 /ATS-VOL1533867958
p ATS-VOL1533867958 default 2049.138.131538 -> 2322.16.2  physical20.qa.lab:5660 physical19.qa.lab:5660 physical
22.qa.lab:5660
vrwxr-xr-x  Z E U U   3 root root          1 2018-08-09 21:31 268435456 /ATS-VOL1533875473
p ATS-VOL1533875473 default 2049.190.131642 -> 2685.16.2  physical21.qa.lab:5660 physical27.qa.lab:5660 physical
23.qa.lab:5660
drwxr-xr-x  Z E U U   - root root          1 2018-08-09 18:15 268435456 /ATS-VOLUME-1533863729955
p 2049.102.131466  physical19.qa.lab:5660 physical20.qa.lab:5660 physical23.qa.lab:5660
...
```

In the above example, the letter **Z** indicates LZ4 compression on the directory; the letter **U** indicates that the directory is uncompressed. In the following example, the listed item is both uncompressed (first **U**) and unencrypted (second **U**).

```
[root@node1-302 ~]# hadoop mfs -ls /hbase
Found 10 items
drwxr-xr-x  Z E U U   - root root          1 2018-08-09 19:26 268435456 /ATS-VOL1533867958/data1533867963
p 2322.32.131374  physical20.qa.lab:5660 physical19.qa.lab:5660 physical22.qa.lab:5660
...
```

Set ACEs

Example 1: The following command shows how to set separate read, write, and execute permissions (using ACE) on a file:

```
hadoop mfs -setace -readfile p -writefile 'g:group1&!u:user1' -execute file p /file
```

When the command shown above runs, the POSIX mode bits for:

- Read access is set for owner, owning group, and others.
- Write access is set for none.
- Execute access is set for owner, owning group, others.

All other POSIX mode bits are set to 0.

Example 2: The following command shows how to set read, write, and execute permissions (using ACE) as a single string on the specified directory and force all files and subdirectories under the specified directory to inherit the parent ACE. ACEs that are not specified will be set to the empty string.

```
hadoop mfs -setace -aces "rf:u:root,wf:group1&!user1,ef:p,rd:u:m7user1" -setinherit true /dir
```

When the command shown above runs, the POSIX mode bits for listing the contents (**r**) of the directory is set for owner/user and all other POSIX mode bits on the directory are set to 0; all new directories under this directory will inherit the parent POSIX mode bits. Also, new files in the directory will inherit the following POSIX mode bits:

- Read access is set to owner/user.
- Write access is set to none.
- Execute access set for others.

All other POSIX mode bits are set to 0.

Example 3: The following command shows how to set permissions (using ACE) as a single string on the specified directory and all the files and subdirectories recursively.

```
hadoop mfs -setace -R -aces "rf:p,wf:g:group1&!u:user1,ef:p" -preservemodebits true /dir
```

When the command shown above runs, the POSIX mode bits are not modified to match the ACE setting.

Example 4: The following command shows how to deny a specific type of access, **writefile**, which was set in the first example above, without removing all other access types associated with the file. The empty string used in the following example will deny write access to all users, roles, and groups.

```
hadoop mfs -setace -writefile "" -preservemodebits false /file
```

When the command shown above runs, the POSIX mode bit for writing to the file is set to 0.

Get ACEs

The following command shows the ACEs and POSIX mode bits for the specified file only.

```
hadoop mfs -getace /m7user1/file1.txt
```

Output

```
Path: /m7user1/file1.txt
readfile: !u:m7user1
writefile: !u:m7user1
executefile: !u:m7user1
mode: -----
```

Delete ACEs

The following command deletes all ACEs associated with the specified file and sets the ACE for the file to the empty string.

```
hadoop mfs -delace /file
```

The following command deletes all ACEs associated with the specified directory.

```
hadoop mfs -delace /dir
```

The following command deletes all ACEs associated with the specified directory and ACEs associated with the files and directories (recursively) below the specified directory.

```
hadoop mfs -delace -R /dir
```

Create a Hard Link to File

The following command shows how to create a hard link to the file, file1, using a new name, file2.

```
# hadoop mfs -lnh /madvol1/file1 /madvol1/file2
Creating Hardlink: /madvol1/file2 -> /madvol1/file1
```

Retrieve the Number of Hard Links

The following command shows how to retrieve the number of hard links (and other statistics) associated with a given file.

```
# hadoop mfs -stat /vol1/file1
Path: /vol1/file1
fid: 23185.32.131232
uid: root
gid: root
atime: 2016-06-29 18:49:03
mtime: 2016-07-01 18:01:54
nlink: 2
type: FTRegular
subtype: FSTInval
size: 1024000000
blocksize: 268435456
mode: 644
networkencryption: false
compression: off
```

View the status of the offload or recall operation for the file named file2 in volume named vol1:

```
# hadoop mfs -tierstatus /vol1/file2
File has local data.
```

View the status of file named test1 in volume named vol1:

```
# hadoop mfs -tierstatus /vol1/test1 -v
      FID          Has Local Data
2154.109.1049824    Yes
2172.143.524906     Yes
2172.153.524926     Yes
2172.166.524952     Yes
2172.167.524954     Yes
File has local data.
```