

# MEGAcmd

## Command Summary

These summaries use the usual conventions - `[]` indicates its content is optional,  
| indicates you should choose either the item on the left or the one on the right (but not both)  
Each command is described as it would be used in the interactive MEGAcmd shell,  
and the corresponding scriptable command (which must be prefixed with `mega-`) works in the same way.  
Commands referring to a remote path are talking about a file in your MEGA account online, whereas a local path refers to a file or folder on your local device where MEGAcmd is running.  
Verbosity: You can increase the amount of information given by any command by passing `-v` (`-vv`, `-vvv`, ...)

## What is it

This document relates to MEGAcmd version 0.9.9. It contains introductory information and the Command Summary, with links to detailed command descriptions.

### What is it

A command line tool to work with your MEGA account and files. The intent is to offer all the MEGA account functionality via command line. You can run it in interactive mode where it processes all commands directly, or you can run its scriptable commands from your favourite Linux or Mac shell such as bash, or you can even run its commands in a Windows command prompt. And of course you can write scripts using those scriptable commands.

Here is an example of downloading a file using MEGAcmd. In this case we are downloading a file specified by a public link, which does not require being logged in:

```
mega-get https://mega.nz/#F!ABcD1E2F!gHiJ23k-LMno45PqrSTUvw /path/to/local/folder
```

And here is an example of uploading a file using MEGAcmd, and making a link available to share it, that will expire after 10 minutes.

```
mega-put /path/to/my/temporary_resource /exportedstuff/
```

```
mega-export -a /exportedstuff/temporary_resource --expire=10M | awk '{print $4}'
```

And here is an example of the power of using scriptable MEGAcmd commands in bash. In this case we are going to share some promotional videos previously uploaded to MEGA:

```
for i in $(mega-find /enterprise/video/promotional2015/may --pattern="*mpeg")
do
mega-export -a $i | awk '{print $4}';
done
```

In addition to running commands on request, MEGAcmd can also be configured to synchronise folders between your local device and your MEGA account, or perform regular backups from your device to your MEGA account.

In order to enable synchronisation and backup features, and for efficiency running commands, MEGAcmd runs a server process in the background which the MEGAcmd shell or the script commands forward requests to. The server keeps running in the background until it is told to close with the `quit` command. If you want it to keep running when you quit the interactive shell (to keep sync and backup running for example), use `quit --only-shell`.

Working with your MEGA account requires signing in with your email and password using the `login` command, though you can download public links or upload to public folders without logging in. Logging in with your username and password starts a Session, and causes some of your account such as the folder structure to be downloaded to your Local Cache.

### Where can you get it

For Linux, Mac, or Windows: Download it from the MEGA.nz website: <https://mega.nz/cmd>

We are also building it for some NAS systems, please check your provider's App Store.

### What can you do with it

The major features are

- Move files around inside your MEGA account or between MEGA and your PC using command line tools.
- Use those same commands in scripts to manage your files.
- Set up synchronization or a backup schedule between a folder on your machine, and a folder on your MEGA account. (use the `sync` or `backup` commands)

- Set up WebDAV access to files in your MEGA account (use the `webdav` command)

See our Help Centre pages for the basics of getting started, and friendly examples of common usages with plenty of pictures: <https://mega.nz/help>

## Terminology and Descriptions

### Interactive

Interactive refers to running the MEGAcmd shell which only processes MEGA commands. You invoke commands by typing and pressing Enter. MEGAcmd shell provides a lot of feedback about what it's doing. You can start the MEGAcmd shell with `mega-cmd` (or MEGAcmd on Windows). You can then issue commands like `ls` directly:

```
ls /my/account/folder
```

or you can get a list of available commands with:

```
help
```

or you can get detailed information about any particular command by using the `--help` flag with that command:

```
ls --help
```

Autocompletion (pressing tab to fill in the remainder of a command) is available in interactive mode.

### Scriptable

Scriptable refers to running the MEGAcmd commands from a shell such as bash or the windows powershell. If the PATH to the MEGAcmd commands are not yet on the PATH in that shell, you'll need to add it. You can then issue commands like `ls` by prefixing them with the `mega-` prefix:

```
mega-ls /my/account/folder
```

or you can get a list of available commands with:

```
mega-help
```

or you can get detailed information about any particular command by using the `--help` flag with that command:

```
mega-ls --help
```

Scriptable commands can of course be used in scripts to achieve a lot in a short space of time, using loops or preparing all the desired commands ahead of time. If you are using bash as your shell, the MEGAcmd commands support auto-completion.

### Contact

A contact is someone (identified by their email address) that also has a MEGA account, who you can share files or folders with, and can chat with on MEGAchats.

### Remote Path

This refers to a file or a folder stored in your MEGA account, or a publicly available file or folder in the MEGA cloud. Remote paths always use the `'/'` character as the separator between folder and file elements.

Some MEGAcmd commands allow the use of regular expressions in remote paths. You can check if the command supports those by using the `--help` flag with the command. If you use these in the scriptable way, you need to escape characters that would otherwise be intercepted and interpreted by the shell.

### Local Path

This refers to a file or folder on the PC or device that MEGAcmd is running in.

### Session

When you log in with your email and MEGA account password, that creates a session. The session exists until you log out of it or kill it from another client. In MEGAcmd, use `whoami -l` to see all your open sessions across all devices, and use `killsession` to close them. You can use other MEGA clients such as the phone app, or webclient to close these also. Devices that were using a killed session will have their connection to MEGA closed immediately and will no longer have access to your account, until you log in on them again. Syncs, backups, and webdavs are specific to a session, so logging out will cause them to be cancelled.

### Local Cache

Logging in with MEGAcmd creates your Local Cache, a subfolder of your home folder. MEGAcmd downloads and stores some data in your Local Cache relating to your account, such as folder structure and contacts, for performance reasons. The MEGAcmd background server

keeps the local cache up to date when changes to your account occur from other clients. The cache does contain a way for MEGAcmd to access your MEGA account when it starts up again if you have not specifically logged out. The Local Cache also contains information from your Session, including sync, backup, and webdav configurations. Logging out cleans the Local Cache, but also closes your session and the sync, backup, and webdav configurations are wiped.

## Synchronisation configurations

MEGAcmd can set up a synchronisation between a folder on your local machine and a folder in your MEGA account, using the `sync` command. This is the same mechanism that MEGAsync uses. The synchronisation is two-way: the folders you nominate to be synced will mirror any action! Whatever you add or delete in your sync folder on your device gets added or deleted in your sync folder in your MEGA account. And additions or deletions in your synced folder in your MEGA account will similarly be applied to your local synced folder. Files that are removed from sync folders are moved to a hidden local folder (Rubbish/.debris inside your local sync folder, or SyncDebris folder in the Rubbish Bin of your MEGA account).

Here is a very simple example of setting up a synchronisation with MEGAcmd:

```
sync /path/to/local/folder /folder/in/mega
```

You can set up more than one pair of folders to be synced, and you can also set a sync from another device to the same folder, to achieve folder synchronisations between different devices. The changes are sent via your MEGA account rather than directly between the devices in that case.

Additional information about synchronising folders is available in our Help Centre: <https://mega.nz/help/client/megasync/syncing>

## Backup configurations

MEGAcmd can set up a periodic copy of a local folder to your MEGA account using the `backup` command. Here is a simple example that will back up a folder immediately and then at 4am each day, keeping the 10 most recent backups:

```
backup /path/mega/folder /remote/path --period="0 0 4 * * *" --num-backups=10
```

For further information on backups, please see the `backup` command and the tutorial.

## WebDAV configurations

MEGAcmd can set up access to folders or files in your MEGA account as if they were local folders and files on your device using the `webdav` command. For example making the folder appear like a local drive on your PC, or providing a hyperlink a browser can access, where the hyperlink is to your PC.

For further information on WebDAV, please see the `webdav` command and the tutorial.

## Linux

On Linux, MEGAcmd commands are installed at `/usr/bin` and so will already be on your PATH. The interactive shell is `mega-cmd` and the background server is `mega-cmd-server`, which will be automatically started on demand. The various scriptable commands are installed at the same location, and invoke `mega-exec` to send the command to `mega-cmd-server`.

If you are using the scriptable commands in bash (or using the interactive commands in `mega-cmd`), the commands will auto-complete.

## Macintosh

For MacOS, after installing the dmg, you can launch the server using MEGAcmd in Applications. If you wish to use the client commands from MacOS Terminal, open the Terminal and include the installation folder in the PATH.

Typically:

```
export PATH=/Applications/MEGAcmd.app/Contents/MacOS:$PATH
```

And for bash completion, source `megacmd_completion.sh`:

```
source /Applications/MEGAcmd.app/Contents/MacOS/megacmd_completion.sh
```

## Windows

Once you have MEGAcmd installed, you can start the interactive shell from the Start Menu or desktop icon. On windows the interactive shell executable is called `MEGAcmdShell.exe` and the server is `MEGAcmdServer.exe`.

On Windows 7, we recommend using the MEGAcmd shell from inside PowerShell for a better user experience (and you can do this on other Windows platforms also). You can start powershell from the Start Menu and then execute these commands to start it:

```
$env:PATH += ";$env:LOCALAPPDATA\MEGAcmd"
```

```
MEGAcmdShell
```

For scriptable usage, the MEGAcmd commands are provided via installed .bat files which pass the command to the MEGAcmdServer.exe. Provided you have set the PATH as above, you can use these like normal command line tools in PowerShell:

```
$env:PATH += ";$env:LOCALAPPDATA\MEGAcmd"
```

```
mega-cd /my/favourite/folder
```

```
mega-ls
```

Or in Command Prompt:

```
set PATH=%LOCALAPPDATA%\MEGAcmd;%PATH%
```

```
mega-cd /my/favourite/folder
```

```
mega-ls
```

And of course those can be invoked in your own .bat or .cmd files. Autocompletion is not available for the scriptable commands, but is in the interactive shell.

Unicode is supported though it currently in the interactive shell it needs to be switched on, and to have a suitable font selected; please execute `help --unicode` for the latest information. There are plans to improve this. Please report any issues experienced to our support team.

## NAS Support

We have released packages for QNAP and Synology, which you can download and install from the App Center in QNAP, and the Package Center in Synology. In QNAP, please make sure to turn on "Enable home folder for all users" from the control panel, and set `HOME=/share/homes/` before starting any MEGA commands, and in Synology, 'Enable user home service', so that the `mega-cmd-server` creates the `.megaCmd` local cache folder there (as the default HOME location may be erased on restart).

## Usage

Before explaining the two ways of interaction, it is important to understand how MEGAcmd works. When you login with MEGAcmd, your session, the list of synced folders, and some cache database are stored in your local home folder. MEGAcmd also stores some other configuration in that folder. Closing it does not delete those and restarting your computer will restore your previous session (the same as `megasync` won't ask for user/password once you restart your computer).

You will need to `logout` properly in order to clean your data.

Now let's get into details of the two usage modes. Both modes require that MEGAcmdServer is running. You can manually launch it. Fortunately, you can also open the interactive shell or execute any command and the server will start automatically.

### Interactively:

Execute MEGAcmd shell. `Platform` section explains how to do that in the different supported systems. You should be facing an interactive shell where you can start typing your commands, with their arguments and flags.

You can list all the available commands with `help`.

And obtain useful information about a command with `command --help`

First you would like to log in into your account. Again, notice that doing this stores the session and other stuff in your home folder. A complete logout is required if you want to end you session permanently and clean any traces (see `logout --help` for further info).

### Non-interactively:

When MEGAcmd server is running, it will be listening for client commands. Use the different `mega-*` commands available.

`mega-help` will list all these commands (you will need to prepend "mega-" to the commands listed there). To obtain further info use

```
mega-command --help
```

Those commands will have an output value `!= 0` in case of failure. See `megacmd.h` to view the existing error codes.

Ideally, you would like to have these commands in your `PATH` variable (See `Platform` for more info). For further info use `mega-help --non-interactive`.

## Usage examples

Here are some examples of use (more info and usage examples are available at the User Guide).

Notice: the commands listed here assume you are using the interactive interaction mode: they are supposed to be executed within MEGAcmdShell.

- A synchronization can be established simply by typing:

```
sync /path/to/local/folder /folder/in/mega
```

This will synchronize the contents in your local and your mega folder both ways.

- You can also set remote backups of a local folder to keep historical snapshots of your files. So simple as:

```
backup /path/mega/folder /remote/path --period="0 0 4 * * *" --num-backups=10
```

This will configure a backup of "myfolder" into /remote/path that will be carried out at 4:00 A.M. (UTC) every day. It will store the last 10 copies.

Further info on backups here.

- You serve a location in your MEGA account via webdav:

```
webdav /path/mega/folder
```

- Or stream a file in your MEGA account:

```
webdav /path/to/myfile.mp4
```

Further info on webdav and streaming here.

- Download the contents of a shared link:

```
get https://mega.nz/#F!ABcD1E2F!gHiJ23k-LMno45PqrSTUvw /path/to/local/folder
```

Now let's do something more complicated with non-interactive usage using some GNU tools (similar stuff can be easily done in Windows as well):

- We want to provide something crypto secured with only 10 minutes of access:

```
mega-put /path/to/my/temporary_resource /exportedstuff/
```

```
mega-export -a /exportedstuff/temporary_resource --expire=10M | awk '{print $4}'
```

- Or imagine we'd like to public the enterprise promotional videos of May 2015 that we have previously stored in MEGA:

```
for i in $(mega-find /enterprise/video/promotional2015/may --pattern="*mpeg"); do
```

```
mega-export -a $i | awk '{print $4}';
```

```
done
```

## Platforms

### Linux

If you have installed MEGAcmd using one of the available packages at [here](#). Or have it built without `--prefix`, both the server (`mega-cmd-server`), the shell (`mega-cmd`) and the different client commands (`mega-*`) will be in your `PATH` (on a fresh install, you might need to open your terminal again).

If you are using bash, you should also have autocompletion for client commands working.

If that is not your case, include the location for the binaries in your `PATH` variable.

### Windows

Once you have MEGAcmd installed, you just need to execute it (via Desktop icon or Start Menu) to open the shell. This will open MEGAcmdServer in the background (a process named MEGAcmdServer.exe).

For a better user experience (specially in Windows 7) we recommend executing MEGAcmd from PowerShell.

Open PowerShell and execute:

```
$env:PATH += ";$env:LOCALAPPDATA\MEGAcmd"
```

```
MEGAcmdShell
```

For non-interactive usage, there are several `mega-*.bat` client commands you can use writing their absolute paths, or including their location into your environment `PATH` and execute them normally (`mega-*`).

If you use PowerShell and you have installed the official MEGAcmd, you can do that simply with:

```
$env:PATH += ";$env:LOCALAPPDATA\MEGAcmd"
```

Client commands completion requires bash, hence, it is not available for Windows.

### Caveats

Although there have been several efforts in having non-ASCII unicode characters supported in Windows, there still may be some issues.

Pay special attention if you are willing to use pipes or send the output of a command into a file from your client commands. See `help --unicode` for further info regarding that.

### MacOS

For MacOS, after installing the dmg, you can launch the server using MEGAcmd in Applications. If you wish to use the client commands from MacOS Terminal, open the Terminal and include the installation folder in the `PATH`. Typically:

```
export PATH=/Applications/MEGAcmd.app/Contents/MacOS:$PATH
```

And for bash completion, source `megacmd_completion.sh`:

```
source /Applications/MEGAcmd.app/Contents/MacOS/megacmd_completion.sh
```

### NAS systems

Currently we have build scripts for Synology and QNAP, which can be found in the `build/` folder along with instructions on how to set up the build. Typically this results in a 'package' which can then be manually installed in the NAS. To use MEGAcmd on those systems, ssh into the device and run the commands as normal (having first added their folder to your `PATH` variable).

## Features:

### Autocompletion:

MEGAcmd features autocompletion in both interactive and non-interactive (only for bash) mode. It will help completing both local and remote (Mega Cloud) files, flags for commands, values for flags/access levels, even contacts.

### Verbosity

There are two different kinds of logging messages:

- SDK based: those messages reported by the sdk and dependent libraries.
- MEGAcmd based: those messages reported by MEGAcmd itself.

You can adjust the level of logging for those kinds with `log` command.

However, passing `-v` (`-vv`, `-vvv`, and so on for a more verbose output) to an specific command will use higher level of verbosity of MEGAcmd based messages.

Further info on verbosity [here](#).

### Regular Expressions

If you have compiled MEGAcmd with PCRE (enabled by default), you can use PCRE compatible expressions in certain commands with the flag `--use-pcre`. Otherwise, if compiled with `c++11`, `c++11` regular expressions will be used.

If none of the above is the case, you can only use wildcards: `"*"` for any number of characters or `"?"` for a single unknown character.

You can check the regular expressions compatibility with `find --help`:

```
find --help
```

```
...
```

Options:

```
--pattern=PATTERN  Pattern to match (Perl Compatible Regular Expressions)
```

Notice: if you use MEGAcmd in non interactive mode, notice that shell pattern will take precedence. You will need to either escape symbols like `*` (`\*`) or surround them between quotes (e.g: `"*.txt"`)

### MEGAcmd Updates

MEGAcmd updates automatically for Windows & MacOS.

For Linux, whenever there is a new update, it will be published in the corresponding repository and your system's updating tool will let you update it.

### Disable automatic updates

You can type `update --auto=OFF` to disable automatic updates. `update --auto=ON` will re-enable them.

If you want to see the state of automatic updates you can use `update --auto=query`. This will inform if automatic updates are enabled or not.

Notice that MEGAcmdServer must be running in order to have automatic updates working.

You can also update manually by typing `update` within MEGAcmd. This will check if there are updates available and proceed to update if affirmative. Whenever MEGAcmd is updated it will be restarted (all open instances of MEGAcmdShell will be restarted too).

Alternatively you can also execute `MEGAcmdUpdater.exe` in Windows or `MEGAcmdUpdater` (located at `/Applications/MEGAcmd.app/Contents/MacOS`) in MacOS.

## Account / Contacts

### signup

Register as user with a given email

Usage: `signup email [password] [--name="Your Name"]`

Options:

--name="Your Name"    Name to register. e.g. "John Smith"

You will receive an email to confirm your account.

Once you have received the email, please proceed to confirm the link included in that email with "confirm".

## confirm

Confirm an account using the link provided after the "signup" process.

Usage: `confirm link email [password]` It requires the email and the password used to obtain the link.

## invite

Invites a contact / deletes an invitation

Usage: `invite [-d|-r] dstemail [--message="MESSAGE"]`

Options:

- d                Deletes invitation
- r                Resends the invitation
- message="MESSAGE" Sends the invitation, including your message.

## showpcr

Shows incoming and outgoing contact requests.

Usage: `showpcr [--in | --out]`

Options:

- in Shows incoming requests
- out Shows outgoing invitations

## ipc

Manages contact incoming invitations.

Usage: `ipc email|handle -a|-d|-i`

Options:

- a Accepts invitation
- d Rejects invitation
- i Ignores invitation [WARNING: do not use unless you know what you are doing]

## users

List contacts

Usage: `users [-s] [-h] [-n] [-d contact@email]`

Options:

- s Show shared folders with listed contacts
- h Show all contacts (hidden, blocked, ...)
- n Show users names
- d contact@email Deletes the specified contact

## userattr

Lists/updates user attributes

Usage: `userattr [-s attribute value|attribute] [--user=user@email]`

Options:

- `-s attribute value` sets an attribute to a value
- `--user=user@email` select the user to query

## passwd

Modifies user password

Usage: `passwd [oldpassword newpassword]`

## masterkey

Shows your master key.

Usage: `masterkey pathtosave`

Getting the master key and keeping it in a secure location enables you to set a new password without data loss. Always keep physical control of your master key (e.g. on a client device, external storage, or print)

## Login / Logout

### login

Log into your MEGA account

Usage: `login [email [password]] | exportedfolderurl#key | session`

You can log in either with email and password, with session ID, or into a folder (an exported/public folder). If logging into a folder indicate url#key

### logout

Closes your session for security or to allow subsequently logging into a different account.

Usage: `logout [--keep-session]`

Options:

- `--keep-session` The current session is not closed, allowing logging back into it later using the session ID rather than email/password.

MEGACmd will still log back into your account automatically on restart if you specify `--keep-session`, similar to exiting it without logging out.

## whoami

Print account information

Usage: `whoami [-l]`



Options:

- l Show extended info: total storage used, storage per main folder (see mount), pro level, account balance, and also the active sessions

## Session

When you log in with your email and MEGA account password, that creates a session.

The session exists until you log out of it or kill it from another client.

In MEGAcmd, use `whoami -l` to see all your open sessions across all devices, and use `killsession` to close them.

You can use other MEGA clients such as the phone app, or webclient to close these also.

Devices that were using a killed session will have their connection to MEGA closed immediately and will no longer have access to your account, until you log in on them again. Syncs, backups, and webdavs are specific to a session, so logging out will cause them to be cancelled.

## killsession

Kills a session of current user.

Usage: `killsession [-a|sessionid]`

Options:

- a kills all sessions except the current one

To see all sessions use "whoami -l"

## Browse

### cd

Changes the current remote folder

Usage: `cd [remotepath]`

If no folder is provided, it will be changed to the root folder

### lcd

Changes the current local folder for the interactive console

Usage: `lcd [localpath]`

It will be used for uploads and downloads

If not using interactive console, the current local folder will be that of the shell executing mega commands

### ls

Lists files in a remote path

remotepath can be a pattern (it accepts wildcards: ? and \*. e.g.: f\*00?.txt)

Also, constructions like /PATTERN1/PATTERN2/PATTERN3 are allowed

Usage: `ls [-halRr] [--versions] [remotepath]`

Options:

- R|-r list folders recursively
- l print summary
- SUMMARY contents:
  - FLAGS: Indicate type/status of an element:
    - xxxx
    - |||+---- Sharing status: (s)hared, (i)n share or not shared(-)
    - ||+----- if exported, whether it is (p)ermanent or (t)emporal
    - |+----- e/- wheter node is (e)xported
    - +----- Type(d=folder,-=file,r=root,i=inbox,b=rubbish,x=unsupported)
  - VERS: Number of versions in a file
  - SIZE: Size of the file in bytes:
  - DATE: Modification date for files and creation date for folders:
  - NAME: name of the node
- h Show human readable sizes in summary
- a include extra information
- versions show historical versions

You can delete all versions of a file with "deleteversions"

## pwd

Prints the current remote folder

Usage: `pwd`

## lpwd

Prints the current local folder for the interactive console

Usage: `lpwd`

It will be used for uploads and downloads

If not using interactive console, the current local folder will be that of the shell executing mega comands

## attr

Lists/updates node attributes

Usage: `attr remotepath [-s attribute value|-d attribute]`

Options:

- s attribute value sets an attribute to a value
- d attribute removes the attribute

## du

Prints size used by files/folders

Usage: `du [-h] [--versions] [remotepath remotepath2 remotepath3 ... ]`

remotepath can be a pattern (it accepts wildcards: ? and \*. e.g.: `f*00?.txt`)

Options:

- h Human readable
- versions Calculate size including all versions.

You can remove all versions with ``deleteversions`` and list them with ``ls --versions``

# find

Find nodes matching a pattern

Usage: `find [remotepath] [-l] [--pattern=PATTERN] [--mtime=TIMECONSTRAIN] [--size=SIZECONSTRAIN]`

Options:

- `-l` Prints file info
- `--pattern=PATTERN` Pattern to match (it accepts wildcards: ? and \*. e.g.: `f*00?.txt`)
- `--mtime=TIMECONSTRAIN` Determines time constrains, in the form: `[+-]TIMEVALUE`  
TIMEVALUE may include hours(h), days(d), minutes(M), seconds(s), months(m) or years(y)  
Examples:
  - `"+1m12d3h"` shows files modified before 1 month, 12 days and 3 hours the current moment
  - `"-3h"` shows files modified within the last 3 hours
  - `"-3d+1h"` shows files modified in the last 3 days prior to the last hour
- `--size=SIZECONSTRAIN` Determines size constrains, in the form: `[+-]TIMEVALUE`  
TIMEVALUE may include (B)ytes, (K)ilobytes, (M)egabytes, (G)igabytes & (T)erabytes  
Examples:
  - `"+1m12k3B"` shows files bigger than 1 Mega, 12 Kbytes and 3Bytes
  - `"-3M"` shows files smaller than 3 Megabytes
  - `"-4M+100K"` shows files smaller than 4 Mbytes and bigger than 100 Kbytes

# mount

Lists all the main nodes

Usage: `mount`

# Moving/Copying Files

## mkdir

Creates a directory or a directories hierarchy

Usage: `mkdir [-p] remotepath`

Options:

- `-p` Allow recursive

## cp

Moves a file/folder into a new location (all remotes)

Usage: `cp srcremotepath dstremotepath|dstemail:`

If the location exists and is a folder, the source will be copied there.

If the location doesn't exist, the file/folder will be renamed to the destination name given.

If `"dstemail:"` provided, the file/folder will be sent to that user's inbox (//in)

e.g: `cp /path/to/file user@doma.in:`

Remember the trailing `:"`, otherwise a file with the name of that user (`"user@doma.in"`) will be created

## put

Uploads files/folders to a remote folder

**Usage:** put [-c] [-q] [--ignore-quota-warn] localfile [localfile2 localfile3 ...] [dstremotepath]

Options:

- c Creates remote folder destination in case of not existing.
- q queue upload: execute in the background. Don't wait for it to end'
- ignore-quota-warn ignore quota surpassing warning.  
The upload will be attempted anyway.

Notice that the dstremotepath can only be omitted when only one local path is provided.  
In such case, the current remote working dir will be the destination for the upload.  
Mind that using wildcards for local paths will result in multiple paths.

## get

Downloads a remote file/folder or a public link

**Usage:** get [-m] [-q] [--ignore-quota-warn] exportedlink#key|remotepath [localpath]

If the remotepath is a file, it will be downloaded to folder specified in localpath (or to the current folder if not specified).  
If the localpath (destination) already exists and is the same (by content) then nothing will be done. If it differs, it will create a new file appending " (NUM)".  
If the remotepath or exportedlink is a folder, the folder and its entire contents will be downloaded into the destination folder.

Options:

- q queue download: execute in the background.
- m if the folder already exists, the contents will be merged with the downloaded one (preserving the existing files)
- ignore-quota-warn ignore quota surpassing warning. The download will be attempted anyway.

## preview

To download/upload the preview of a file.

**Usage:** preview [-s] remotepath localpath

If no -s is indicated, it will download the preview.

Options:

- s Sets the preview to the specified file

## thumbnail

o download/upload the thumbnail of a file.

**Usage:** thumbnail [-s] remotepath localpath

If no -s is indicated, it will download the thumbnail.

Options:

- s Sets the thumbnail to the specified file

## mv

Copies files/folders to a new location in your MEGA account

**Usage:** mv srcremotepath [srcremotepath2 srcremotepath3 ..] dstremotepath

If the destination remote path exists and is a folder, the source will be copied there.  
If the destination remote path doesn't exist, the source will be renamed to the given dstremotepath leaf name.

## rm

Deletes a remote file/folder

Usage: `rm [-r] [-f] remotepath`

Options:

- r Delete recursively (for folders)
- f Force (no asking)

## transfers

List or operate with queued transfers

Usage: `transfers [-c TAG|-a] | [-r TAG|-a] | [-p TAG|-a] [--only-downloads | --only-uploads] [SHOWOPTIONS]`

If executed without option it will list the first 10 transfers

Options:

- c (TAG|-a) Cancel transfer with TAG (or all with -a)
- p (TAG|-a) Pause transfer with TAG (or all with -a)
- r (TAG|-a) Resume transfer with TAG (or all with -a)
- only-uploads Show/Operate only upload transfers
- only-downloads Show/Operate only download transfers

Show options:

- show-syncs Show synchronization transfers
- show-completed Show completed transfers
- only-completed Show only completed download
- limit=N Show only first N transfers
- path-display-size=N Use a fixed size of N characters for paths

## transfers example

eg.email@example.co.nz:/tmp-test/Mega.dir\$ **transfers**

| DIR/SYNC TAG                       | SOURCEPATH                          | DESTINYPATH                       | PROGRESS              | STATE  |
|------------------------------------|-------------------------------------|-----------------------------------|-----------------------|--------|
| U 17361                            | \\?\C:\Users\ME\...ebug\megaapi.obj | /tmp-test/Mega.dir/Mega.dir/Debug | 100.00% of 2016.62 KB | ACTIVE |
| U 17362                            | \\?\C:\Users\ME\...megaapi_impl.obj | /tmp-test/Mega.dir/Mega.dir/Debug | 13.64% of 13.85 MB    | ACTIVE |
| U 17363                            | \\?\C:\Users\ME\...g\megaclient.obj | /tmp-test/Mega.dir/Mega.dir/Debug | 0.00% of 15.46 MB     | QUEUED |
| U 17364                            | \\?\C:\Users\ME\..._http_parser.obj | /tmp-test/Mega.dir/Mega.dir/Debug | 0.00% of 85.15 KB     | QUEUED |
| U 17365                            | \\?\C:\Users\ME\...ega_utf8proc.obj | /tmp-test/Mega.dir/Mega.dir/Debug | 0.00% of 312.44 KB    | QUEUED |
| U 17366                            | \\?\C:\Users\ME\...mega_zxcvbn.obj  | /tmp-test/Mega.dir/Mega.dir/Debug | 0.00% of 589.88 KB    | QUEUED |
| U 17367                            | \\?\C:\Users\ME\...ir\Debug\net.obj | /tmp-test/Mega.dir/Mega.dir/Debug | 0.00% of 3.20 MB      | QUEUED |
| U 17368                            | \\?\C:\Users\ME\...r\Debug\node.obj | /tmp-test/Mega.dir/Mega.dir/Debug | 0.00% of 3.73 MB      | QUEUED |
| U 17369                            | \\?\C:\Users\ME\...ntactrequest.obj | /tmp-test/Mega.dir/Mega.dir/Debug | 0.00% of 352.22 KB    | QUEUED |
| U 17370                            | \\?\C:\Users\ME\...Debug\proxy.obj  | /tmp-test/Mega.dir/Mega.dir/Debug | 0.00% of 203.57 KB    | QUEUED |
| ... Showing first 10 transfers ... |                                     |                                   |                       |        |

eg.email@example.co.nz:/tmp-test/Mega.dir\$ **transfers -p 17367**

Transfer 17367 paused successfully.

eg.email@example.co.nz:/tmp-test/Mega.dir\$ **transfers -c 17370**

Transfer 17370 cancelled successfully.

eg.email@example.co.nz:/tmp-test/Mega.dir\$ **transfers**

| DIR/SYNC TAG | SOURCEPATH                          | DESTINYPATH                       | PROGRESS           | STATE  |
|--------------|-------------------------------------|-----------------------------------|--------------------|--------|
| U 17362      | \\?\C:\Users\ME\...megaapi_impl.obj | /tmp-test/Mega.dir/Mega.dir/Debug | 96.32% of 13.85 MB | ACTIVE |
| U 17363      | \\?\C:\Users\ME\...g\megaclient.obj | /tmp-test/Mega.dir/Mega.dir/Debug | 0.20% of 15.46 MB  | ACTIVE |
| U 17364      | \\?\C:\Users\ME\..._http_parser.obj | /tmp-test/Mega.dir/Mega.dir/Debug | 0.00% of 85.15 KB  | QUEUED |
| U 17365      | \\?\C:\Users\ME\...ega_utf8proc.obj | /tmp-test/Mega.dir/Mega.dir/Debug | 0.00% of 312.44 KB | QUEUED |
| U 17366      | \\?\C:\Users\ME\...mega_zxcvbn.obj  | /tmp-test/Mega.dir/Mega.dir/Debug | 0.00% of 589.88 KB | QUEUED |
| U 17367      | \\?\C:\Users\ME\...ir\Debug\net.obj | /tmp-test/Mega.dir/Mega.dir/Debug | 0.00% of 3.20 MB   | PAUSED |
| U 17368      | \\?\C:\Users\ME\...r\Debug\node.obj | /tmp-test/Mega.dir/Mega.dir/Debug | 0.00% of 3.73 MB   | QUEUED |
| U 17369      | \\?\C:\Users\ME\...ntactrequest.obj | /tmp-test/Mega.dir/Mega.dir/Debug | 0.00% of 352.22 KB | QUEUED |
| U 17371      | \\?\C:\Users\ME\...pubkeyaction.obj | /tmp-test/Mega.dir/Mega.dir/Debug | 0.00% of 355.75 KB | QUEUED |

```
U 17372 \\?\C:\Users\ME\...ebug\request.obj /tmp-test/Mega.dir/Mega.dir/Debug 0.00% of 933.14 KB QUEUED
... Showing first 10 transfers ...
eg.email@example.co.nz:/tmp-test/Mega.dir$
```

## speedlimit

Displays/modifies upload/download rate limits

Usage: speedlimit [-u|-d] [-h] [NEWLIMIT]

NEWLIMIT establish the new limit in size per second (0 = no limit)

NEWLIMIT may include (B)ytes, (K)ilobytes, (M)egabytes, (G)igabytes & (T)erabytes.

Examples: "1m12k3B" "3M". If no units are given, bytes are assumed.

Options:

- d Download speed limit
- u Upload speed limit
- h Human readable

Notice: these limits are saved for the next time you execute MEGAcmd server. They will be removed if you logout.

## sync

Sets up synchronisation between a local folder and one in your MEGA account. (example)

Usage: sync [localpath dstremotepath] [-dsr] [ID|localpath]

If no argument is provided, it lists current configured synchronizations

If provided local and remote paths, it will start synchronizing a local folder into a remote folder

If an ID/local path is provided, it will list such synchronization unless an option is specified.

Options:

- d ID|localpath deletes a synchronization
- s ID|localpath stops(pauses) a synchronization
- r ID|localpath resumes a synchronization
- path-display-size=N Use a fixed size of N characters for paths

Syncs are associated with your Session, so logging out will cancel them.

## sync example

```
email_1@example.co.nz:/$ sync c:\Go go-backup/
```

```
email_1@example.co.nz:/$ sync
```

| ID | LOCALPATH | REMOTEPATH | ActState | SyncState | SIZE      | FILES | DIRS |
|----|-----------|------------|----------|-----------|-----------|-------|------|
| 0  | \\?\c:\Go | /go-backup | InitScan | Syncing   | 119.13 KB | 10    | 97   |

```
email_1@example.co.nz:/$ sync
```

| ID | LOCALPATH | REMOTEPATH | ActState | SyncState | SIZE     | FILES | DIRS |
|----|-----------|------------|----------|-----------|----------|-------|------|
| 0  | \\?\c:\Go | /go-backup | InitScan | Syncing   | 61.22 MB | 1252  | 463  |

```
email_1@example.co.nz:/$ sync
```

| ID | LOCALPATH | REMOTEPATH | ActState | SyncState | SIZE      | FILES | DIRS |
|----|-----------|------------|----------|-----------|-----------|-------|------|
| 0  | \\?\c:\Go | /go-backup | InitScan | Syncing   | 232.94 MB | 4942  | 773  |

```
email_1@example.co.nz:/$ sync
```

| ID | LOCALPATH | REMOTEPATH | ActState | SyncState | SIZE      | FILES | DIRS |
|----|-----------|------------|----------|-----------|-----------|-------|------|
| 0  | \\?\c:\Go | /go-backup | Active   | Synced    | 285.91 MB | 7710  | 1003 |

[then on a windows cmd prompt]

```
C:\Users\ME>rmdir /s c:\go\blog
```

```
c:\go\blog, Are you sure (Y/N)? Y
```

[back in MEGAcmd- update has been applied to MEGA already]

```
email_1@example.co.nz:/$ sync
```

| ID | LOCALPATH | REMOTEPATH | ActState | SyncState | SIZE | FILES | DIRS |
|----|-----------|------------|----------|-----------|------|-------|------|
|----|-----------|------------|----------|-----------|------|-------|------|

## exclude

Manages exclusions in syncs.

**Usage:** `exclude [(-a|-d) pattern1 pattern2 pattern3 [--restart-syncs]]`

Options:

- a pattern1 pattern2 ... adds pattern(s) to the exclusion list  
(\* and ? wildcards allowed)
- d pattern1 pattern2 ... deletes pattern(s) from the exclusion list
- restart-syncs Try to restart synchronizations.

Changes will not be applied immediately to actions being performed in active syncs.

After adding/deleting patterns, you might want to:

- a) disable/reenable synchronizations manually
- b) restart MEGAcmd server
- c) use --restart-syncs flag. Caveats:

This will cause active transfers to be restarted

In certain cases --restart-syncs might be unable to re-enable a synchronization.

In such case, you will need to manually resume it or restart MEGAcmd server.

## backup

Sets up or controls backups.

**Usage:** `backup (localpath remotepath --period="PERIODSTRING" --num-backups=N | [-lhda] [TAG|localpath] [--period="PERIODSTRING"] [--num-backups=N])`

This command can be used to configure which folders to back up, and how often to do so.

If no argument is given it will list the configured backups.

To get extra info on backups use -l or -h (see Options below)

When a backup of a folder (localfolder) is established in a remote folder (remotepath)

MEGAcmd will create subfolder within the remote path with names like: "localfoldername\_bk\_TIME"

which shall contain a backup of the local folder at that specific time

In order to configure a backup you need to specify the local and remote paths, the period and max number of backups to store (see Configuration Options below).

Once configured, you can see extended info asociated to the backup (See Display Options)

Notice that MEGAcmd server need to be running for backups to be created.

Display Options:

-l Show extended info: period, max number, next scheduled backup or the status of current/last backup

-h Show history of created backups

Backup states:

While a backup is being performed, the backup will be considered and labeled as ONGOING

If a transfer is cancelled or fails, the backup will be considered INCOMPLETE

If a backup is aborted (see -a), all the transfers will be canceled and the backup be ABORTED

If MEGAcmd server stops during a transfer, it will be considered MISCARRIED

Notice that currently when MEGAcmd server is restarted, ongoing and scheduled transfers will be carried out nevertheless.

If MEGAcmd server is not running when a backup is scheduled and the time for the next one has already arrived, an empty BACKUP will be created with state SKIPPED

If a backup(1) is ONGOING and the time for the next backup(2) arrives, it won't start untill the previous one(1)

is completed, and if by the time the first one(1) ends the time for the next one(3) has already arrived,

an empty BACKUP(2) will be created with state SKIPPED

--path-display-size=N Use a fixed size of N characters for paths

Configuration Options:

--period="PERIODSTRING" Period: either time in TIMEFORMAT (see below) or a cron like expresions

Cron like period is formatted as follows

- - - - -

```

| | | | |
| | | | |
| | | | +---- Day of the Week (range: 1-7, 1 standing for Monday)
| | | | +----- Month of the Year (range: 1-12)
| | | | +----- Day of the Month (range: 1-31)
| | | +----- Hour (range: 0-23)
| | +----- Minute (range: 0-59)
| +----- Second (range: 0-59)
+-----

```

examples:

```

- daily at 04:00:00 (UTC): "0 0 4 * * *"
- every 15th day at 00:00:00 (UTC) "0 0 0 15 * *"
- mondays at 04.30.00 (UTC): "0 30 4 * * 1"

```

TIMEFORMAT can be expressed in hours(h), days(d), minutes(M), seconds(s), months(m) or years(y)

e.g. "1m12d3h" indicates 1 month, 12 days and 3 hours

Notice that this is an uncertain measure since not all months

last the same and Daylight saving time changes are not considered

If possible use a cron like expresion

Notice: regardless of the period expresion, the first time you establish a backup, it will be created immediately

--num-backups=N Maximum number of backups to store

After creating the backup (N+1) the oldest one will be deleted

That might not be true in case there are incomplete backups:

in order not to lose data, at least one COMPLETE backup will be kept

Use backup TAG|localpath --option=VALUE to modify existing backups

Management Options:

-d TAG|localpath Removes a backup by its TAG or local path

Folders created by backup won't be deleted

-a TAG|localpath Aborts ongoing backup

Syncs are associated with your Session, so logging out will cancel them.

Caveat: This functionality is in BETA state. If you experience any issue with this, please contact: support@mega.nz

## backup example

eg.email@example.co.nz:/\$ **backup c:/cmake /cmake-backup --period="0 0 4 \* \* \*" --num-backups=3**

Backup established: c:/cmake into /cmake-backup period="0 0 4 \* \* \*" Number-of-Backups=3

eg.email@example.co.nz:/\$ **backup**

| TAG | LOCALPATH    | REMOTE PARENT PATH | STATUS  |
|-----|--------------|--------------------|---------|
| 166 | \\?\c:\cmake | /cmake-backup      | ONGOING |

eg.email@example.co.nz:/\$ **backup -h**

| TAG | LOCALPATH    | REMOTE PARENT PATH | STATUS  |
|-----|--------------|--------------------|---------|
| 166 | \\?\c:\cmake | /cmake-backup      | ONGOING |

-- SAVED BACKUPS --

| NAME                    | DATE               | STATUS  | FILES | FOLDERS |
|-------------------------|--------------------|---------|-------|---------|
| cmake_bk_20180426133300 | 26Apr2018 13:33:00 | ONGOING | 0     | 92      |

eg.email@example.co.nz:/\$

## MEGA-BACKUPS - Backing folders with MEGAcmd

This is a brief tutorial on how to configure backups.

Notice: the commands listed here assume you are using the interactive interaction mode: they are supposed to be executed within MEGAcmdShell.

### Creation

Example: backup /path/mega/folder /remote/path --period="0 0 4 \* \* \*" --num-backups=10

This will configure a backup of "myfolder" into /remote/path that will be carried out at 4:00 A.M. (UTC) every day. It will store the last 10 copies. Notice a first backup will be carried out immediately. In this example we are using cron-time expresions. You can find extra info on those using "backup --help".

Backups will be stored as:

```

/remotepath/myfolder_bk_TIME1
/remotepath/myfolder_bk_TIME2
/remotepath/myfolder_bk_TIME3
...

```



## Listing

You can list the backups configured typing `backup`:

| TAG | LOCALPATH         | REMOTE PARENT PATH | STATUS |
|-----|-------------------|--------------------|--------|
| 4   | /path/mega/folder | /remote/path       | ACTIVE |

Notice the TAG. You can use it to refer to the backup if you want to change its configuration or delete/abort it.

## Extra info

If you type "`backup -l`" you will see extra information concerning the backup. Here, you will see when the next backup is scheduled for:

| TAG | LOCALPATH         | REMOTE PARENT PATH | STATUS  |
|-----|-------------------|--------------------|---------|
| 4   | /path/mega/folder | /remote/path       | ONGOING |

Max Backups: 4  
Period: "0 0 4 \* \* \*"  
Next backup scheduled for: Fri, 19 Jan 2018 04:00:00 +0000  
-- CURRENT/LAST BACKUP --

| FILES UP/TOT | FOLDERS CREATED | PROGRESS             |
|--------------|-----------------|----------------------|
| 22/33        | 0               | 57.86/57.86 KB 1.27% |

Also, you can see the progress of the current backup (or the last one if there is no backup being performed at the moment)

## Backup history:

With "`backup -h`" you will be able to see the existing backups with their state and start time:

| TAG | LOCALPATH         | REMOTE PARENT PATH | STATUS  |
|-----|-------------------|--------------------|---------|
| 4   | /path/mega/folder | /remote/path       | ONGOING |

-- SAVED BACKUPS --

| NAME                       | DATE               | STATUS   | FILES | FOLDERS |
|----------------------------|--------------------|----------|-------|---------|
| myfolder_bk_20180115175811 | 15Jan2018 17:58:11 | COMPLETE | 33    | 10      |
| myfolder_bk_20180116182611 | 16Jan2018 18:26:11 | COMPLETE | 33    | 10      |
| myfolder_bk_20180117182711 | 17Jan2018 18:27:11 | ABORTED  | 13    | 10      |
| myfolder_bk_20180118182911 | 18Jan2018 18:29:11 | ONGOING  | 23    | 10      |

Tip: If you are using linux/mac you can monitor the status actively in non-interactive mode with:

```
watch mega-backup -lh
```

## Control:

### Abort

You can abort an ONGOING backup using its tag or its local path. e.g.:

```
backup -a 4
```

This will cancel all transfers and set the backup as ABORTED

### Delete

Similarly you can remove a backup, to no longer backup that folder with:

```
backup -d /path/mega/folder
```

This will not remove the existing backups which will be available in MEGA.

## Change configuration

Similarly you can change the period or the number of backups to keep with:

```
backup 4 --period=2h
```

This will set our backup with TAG=4 to have a period of 2 hours.

```
backup /path/mega/folder --num-backups=1
```

This will configure the backup to only keep one instance. Notice that in order not to lose data, older instances will not be deleted until the max number of backups is passed.

# Sharing

## export

Prints/Modifies the status of current exports

Usage: `export [-d|-a [--expire=TIMEDELAY] [-f]] [remotepath]`

Options:

- a Adds an export (or modifies it if existing)
- expire=TIMEDELAY Determines the expiration time of a node.  
It indicates the delay in hours(h), days(d), minutes(M), seconds(s), months(m) or years(y)  
e.g. "1m12d3h" establish an expiration time 1 month, 12 days and 3 hours after the current moment
- f Implicitly accept copyright terms (only shown the first time an export is made)  
MEGA respects the copyrights of others and requires that users of the MEGA cloud service comply with the laws of copyright.  
You are strictly prohibited from using the MEGA cloud service to infringe copyrights.  
You may not upload, download, store, share, display, stream, distribute, email, link to, transmit or otherwise make available any files, data or content that infringes any copyright or other proprietary rights of any person or entity.
- d Deletes an export

If a remote path is given it'll be used to add/delete or in case of no option selected, it will display all the exports existing in the tree of that path

## import

Imports the contents of a remote link into your MEGA account or to a local folder.

Usage: `import exportedfilelink#key [remotepath]`

If no remote path is provided, the current local folder will be used

## share

Prints/Modifies the status of current shares

Usage: `share [-p] [-d|-a --with=user@email.com [--level=LEVEL]] [remotepath]`

Options:

- p Show pending shares
- with=email Determines the email of the user to [no longer] share with
- d Stop sharing with the selected user
- a Adds a share (or modifies it if existing)
- level=LEVEL Level of acces given to the user
  - 0: Read access
  - 1: Read and write
  - 2: Full access
  - 3: Owner access

If a remote path is given it'll be used to add/delete or in case of no option selected, it will display all the shares existing in the tree of that path

When sharing a folder with a user that is not a contact (see "users --help") the share will be in a pending state. You can list pending shares with ``share -p``. Your contact will need to accept your invitation (see [``ipc``](#ipc))

If someone has shared something with you, it will be listed as a root folder.  
Use [``mount``](#mount) to list folders shared with you

## webdav

Configures a WEBDAV server to serve a location in MEGA. You can use `feature` to make a folder in your MEGA account appear as a virtual drive, or to stream files.

**Usage:** `webdav [ [-d] remotepath [--port=PORT] [--public] [--tls --certificate=/path/to/certificate.pem --key=/path/to/certificate.key]]`

This can also be used for streaming files. The server will be running as long as MEGAcmd Server is. If no argument is given, it will list the webdav enabled locations.

Options:

- `--d` Stops serving that location
- `--public` \*Allow access from outside localhost
- `--port=PORT` \*Port to serve. DEFAULT= 4443
- `--tls` \*Serve with TLS (HTTPS)
- `--certificate=/path/to/certificate.pem` \*Path to PEM formatted certificate
- `--key=/path/to/certificate.key` \*Path to PEM formatted key

\*If you serve more than one location, these parameters will be ignored and use those of the first location served.

Webdav setup is associated with your Session, so logging out will cancel them.

Caveat: This functionality is in BETA state. If you experience any issue with this, please contact: [support@mega.nz](mailto:support@mega.nz)

## webdav example

```
eg.email@example.co.nz:/$ webdav myfile.tif --port=1024
Serving via webdav myfile.tif: http://127.0.0.1:1024/5mYHQT4B/myfile.tif
eg.email@example.co.nz:/$ webdav
WEBDAV SERVED LOCATIONS:
/myfile.tif: http://127.0.0.1:1024/5mYHQT4B/myfile.tif
eg.email@example.co.nz:/$ webdav -d myfile.tif
myfile.tif no longer served via webdav
eg.email@example.co.nz:/$
```

## Misc

### version

Prints MEGAcmd versioning and extra info

**Usage:** `version [-l] [-c]`

Options:

- `-c` Shows changelog for the current version
- `-l` Show extended info: MEGA SDK version and features enabled

## deleteversions

Deletes previous versions of files, keeping the current version.

**Usage:** `deleteversions [-f] (--all | remotepath1 remotepath2 ...)`

This will permanently delete all historical versions of a file.

The current version of the file will remain.

Note: any file version shared to you from a contact will need to be deleted by them.

Options:

-f Force (no asking)

--all Delete versions of all nodes. This will delete the version histories of all files (not current files).

To see versions of a file use `ls --versions`.

To see space occupied by file versions use `du --versions`.

## unicode

Toggle unicode input enabled/disabled in interactive shell

Usage: `unicode`

Unicode mode is experimental, you might experience some issues interacting with the console (e.g. history navigation fails).

Type "help --unicode" for further info.

## reload

Forces re-downloading the Local Cache information from MEGA.

Usage: `reload`

It will also resume synchronizations.

## help

Prints list of commands

Usage: `help [-f]`

Options:

-f Include a brief description of the commands

## https

Shows if HTTPS is used for transfers. Use `https on` to enable it.

Usage: `https [on|off]`

HTTPS is not necessary since all data is already encrypted before being stored or transfered anyway. Enabling it will increase CPU usage and add network overhead.

This setting is ephemeral: it will reset for the next time you open MEGAcmd

## clear

Clear screen

Usage: `clear`

## log

Prints/Modifies the setting for how detailed log output is.

Usage: log [-sc] level

Options:

- c CMD log level (higher level messages).  
Messages captured by MEGAcmd server.
- s SDK log level (lower level messages).  
Messages captured by the engine and libs

Regardless of the log level of the interactive shell, you can increase the amount of information given by any command by passing ``-v`` (``-vv``, ``-vvv``, ...)

## debug

Enters debugging mode (HIGHLY VERBOSE)

Usage: debug

For a finer control of log level see [``log``](#log)

## exit

Quits MEGAcmd

Usage: exit [--only-shell]

By default this command will exit both the interactive shell and the command server.

To only exit current shell and keep server running, use ``exit --only-shell``

Exiting the server does not cancel the session, and the encrypted local cache of your account is kept on your PC. The session will be resumed when the service is restarted.

## Example

### signup confirm invite showpcr ipc users userattr

signup confirm invite showpcr ipc users userattr example

MEGA CMD> **signup eg.email\_1@example.co.nz --name="test1"**

New Password:

Retype New Password:

Account created successfully. You will receive a confirmation link. Use "confirm" with the provided link to confirm that account

MEGA CMD> **confirm https://mega.nz/#confirmQFSfjtUkExc5M2Us6q5d-klx60RfxVbxjkh eg.email\_1@example.co.nz**

Password:

Account eg.email\_1@example.co.nz confirmed successfully. You can login with it now

MEGA CMD> **signup eg.email\_2@example.co.nz --name="test2"**

New Password:

Retype New Password:

Account created successfully. You will receive a confirmation link. Use "confirm" with the provided link to confirm that account

MEGA CMD> **confirm https://mega.nz/#confirmcz7Ss68ChhMKk8WEFTQCqLMHJg8esAEEpQE eg.email\_2@example.co.nz**

Password:

Account eg.email\_2@example.co.nz confirmed successfully. You can login with it now

MEGA CMD> **login eg.email\_1@example.co.nz**

Password:

[API:info: 23:19:14] Fetching nodes ...

[API:info: 23:19:17] Loading transfers from local cache

[API:info: 23:19:17] Login complete as eg.email\_1@example.co.nz

MEGA CMD>

eg.email\_1@example.co.nz:/\$ **invite eg.email\_2@example.co.nz**

Invitation to user: eg.email\_2@example.co.nz sent

```

eg.email_1@example.co.nz:/$ showpcr
Outgoing PCRs:
eg.email_2@example.co.nz (id: 47Xhz6wvVTk, creation: Thu, 26 Apr 2018 11:20:09 +1200, modification: Thu, 26 Apr 2018 11:20:09 +1200)
eg.email_1@example.co.nz:/$ logout
Logging out...
eg.email_1@example.co.nz:/$
MEGA CMD> login eg.email_2@example.co.nz
Password:
[API:info: 23:21:10] Fetching nodes ...
[API:info: 23:21:12] Loading transfers from local cache
[API:info: 23:21:12] Login complete as eg.email_2@example.co.nz
MEGA CMD>
eg.email_2@example.co.nz:/$ showpcr
Incoming PCRs:
eg.email_1@example.co.nz (id: 47Xhz6wvVTk, creation: Thu, 26 Apr 2018 11:20:09 +1200, modification: Thu, 26 Apr 2018 11:20:09 +1200)
eg.email_2@example.co.nz:/$ ipc 47Xhz6wvVTk -a
Accepted invitation by eg.email_1@example.co.nz
eg.email_2@example.co.nz:/$ users
eg.email_1@example.co.nz, visible since Thu, 26 Apr 2018 11:22:02 +1200
eg.email_2@example.co.nz:/$ userattr --user=eg.email_1@example.co.nz
    firstname = test1
    ed25519 = 5Xl2-mUtsZkaATmSS88Ncepju5805uw66Hfdh_-SwpE
    cu25519 = ejoYtpaJIZvlpmpPsYviIa6tNvPTdVjfkYf9G1k8PKgM
    rsa = AAAAAFrhDPPMS1AXAhJwScpJ_GKqFUJ42uIIcwxLp5RIalkWtsa5j87u2LFhoZII_rHIzGXrdsbywgs7Msisw0CjodrtwtME
    cu255 = AAAAAFrhDPPWUOP2tNBvV72zU4M3EKNoddyVCT13VkkouMldniR2UZtLrPjUjUeOZOLvOL7H1C0W0Q_b3QqYSvAKo775pUwD
eg.email_2@example.co.nz:/$ showpcr
eg.email_2@example.co.nz:/$ logout
Logging out...
MEGA CMD> login eg.email_1@example.co.nz
Password:
[API:info: 23:24:26] Fetching nodes ...
[API:info: 23:24:27] Loading transfers from local cache
[API:info: 23:24:27] Login complete as eg.email_1@example.co.nz
MEGA CMD>
eg.email_1@example.co.nz:/$ showpcr
eg.email_1@example.co.nz:/$ users
eg.email_2@example.co.nz, visible
eg.email_1@example.co.nz:/$ userattr --user=eg.email_2@example.co.nz
    firstname = test2
    ed25519 = M7SLy2RajwUAvynxJQaVkh6hxGpbwJmvve3dgl8B1o
    cu25519 = VaXluGS2c5xbo0xOHHJciqLRxwMaWZHVk8iuxtlCBTk
    rsa = AAAAAFrhDWemabQ4JA0tP7zcoy6m74PsFTFCbj04Zh4G8K_TZB5Sm9T5Xj9CXyZwWnpfRd1McPdDouKdsASQ6Er7i4Y4LpEA
    cu255 = AAAAAFrhDWcXE_7AHZmvxk5Hk0G7V65UnvFO42tb1gM9SYy3BpsMCas0X-pbqkYwf6_2eBG-ZLvkonGfXB3DwonWNvnVehIB
eg.email_1@example.co.nz:/$

```

## login logout whoami mkdir cd get put du mount

```

MEGA CMD> login eg.email_1@example.co.nz
Password:
[API:info: 23:43:14] Fetching nodes ...
[API:info: 23:43:14] Loading transfers from local cache
[API:info: 23:43:14] Login complete as eg.email_1@example.co.nz
MEGA CMD>
eg.email_1@example.co.nz:/$ whoami -l
Account e-mail: eg.email_1@example.co.nz
Available storage: 50.00 GBytes
In ROOT: 146... KBytes in 1 file(s) and 0 folder(s)
In INBOX: 0.00 Bytes in 0 file(s) and 0 folder(s)
In RUBBISH: 0.00 Bytes in 0 file(s) and 0 folder(s)
Total size taken up by file versions: 0.00 Bytes
Pro level: 0
Subscription type:
Account balance:
Current Active Sessions:
* Current Session
Session ID: m3a8eluyPdo
Session start: 4/26/2018 11:43:12 AM

```

Most recent activity: 4/26/2018 11:43:13 AM  
IP: 122.56.56.232  
Country: NZ  
User-Agent: MEGAcmd/0.9.9.0 (Windows 10.0.16299) MegaClient/3.3.5  
-----

```
1 active sessions opened
eg.email_1@example.co.nz:/$ mount
ROOT on /
INBOX on //in
RUBBISH on //bin
eg.email_1@example.co.nz:/$ ls
Welcome to MEGA.pdf
eg.email_1@example.co.nz:/$ get "Welcome to MEGA.pdf"
TRANSFERRING ||
#####|(1/1
MB: 100.00 %)
eg.email_1@example.co.nz:/$ mkdir my-pictures
eg.email_1@example.co.nz:/$ cd my-pictures/
eg.email_1@example.co.nz:/my-pictures$ put C:\Users\MYWINDOWSUSER\Pictures
TRANSFERRING ||
#####|(1/1
MB: 100.00 %)
eg.email_1@example.co.nz:/my-pictures$ pwd
/my-pictures
eg.email_1@example.co.nz:/my-pictures$ ls
Pictures
eg.email_1@example.co.nz:/my-pictures$ cd Pictures/
eg.email_1@example.co.nz:/my-pictures/my-pictures$ ls
Camera Roll
Feedback
Saved Pictures
megacmdpkg.gif
megacmdpkg_80.gif
megacmdpkg_gray.gif
eg.email_1@example.co.nz:/my-pictures/my-pictures$ pwd
/my-pictures/Pictures
eg.email_1@example.co.nz:/my-pictures/my-pictures$ cd /
eg.email_1@example.co.nz:/$ du my-pictures/
FILENAME      SIZE
my-pictures:  1376143
-----
Total storage used:      1376143
eg.email_1@example.co.nz:/$ logout
Logging out...
MEGA CMD>
```

## export import

```
eg.email_1@example.co.nz:/$ export -a Pictures/
MEGA respects the copyrights of others and requires that users of the MEGA cloud service comply with the laws of copyright.
You are strictly prohibited from using the MEGA cloud service to infringe copyrights.
You may not upload, download, store, share, display, stream, distribute, email, link to, transmit or otherwise make available any files, data
or content that infringes any copyright or other proprietary rights of any person or entity. Do you accept this terms? (Yes/No): Yes
Please enter [y]es/[n]o/[a]ll/none:yes
Exported /Pictures: https://mega.nz/#F!iaZIEBIL!mQD3rFuJhKov0sco-6s9xg
eg.email_1@example.co.nz:/$ export
Pictures (folder, shared as exported permanent folder link: https://mega.nz/#F!iaZIEBIL!mQD3rFuJhKov0sco-6s9xg)
eg.email_1@example.co.nz:/$ logout --keep-session
Logging out...
Session closed but not deleted. Warning: it will be restored the next time you execute the application. Execute "logout" to delete the
session permanently.
You can also login with the session id: ARo7aiLAXK-jseOdVBYhj285Twb06ivWsFmT4XAnkTsiaDRRbm5oYS1zRm-V3I0FHHOvwj7P2RPvrSw_
MEGA CMD> login eg.email_2@example.co.nz
Password:
[API:info: 01:55:04] Fetching nodes ...
[API:info: 01:55:05] Loading transfers from local cache
[API:info: 01:55:05] Login complete as eg.email_2@example.co.nz
MEGA CMD>
eg.email_2@example.co.nz:/$ ls
```

```
Welcome to MEGA.pdf
eg.email_2@example.co.nz:/$ import https://mega.nz/#F!iaZIEBIL!mQD3rFuJhKov0sco-6s9xg
Imported folder complete: /Pictures
eg.email_2@example.co.nz:/$ ls
Pictures
Welcome to MEGA.pdf
eg.email_2@example.co.nz:/$ ls Pictures/
Camera Roll
Feedback
Saved Pictures
megacmdpkg.gif
megacmdpkg_80.gif
megacmdpkg_gray.gif
eg.email_2@example.co.nz:/$ logout
Logging out...
eg.email_2@example.co.nz:/$
MEGA CMD> login ARo7ailAxK-jseOdVBYhj285Twb06ivWsFmT4XAnkTsiaDRRbm5oYS1zRm-V3I0FHHOvwj7P2RPvrSw_
eg.email_1@example.co.nz:/$ export
Pictures (folder, shared as exported permanent folder link: https://mega.nz/#F!iaZIEBIL!mQD3rFuJhKov0sco-6s9xg)
eg.email_1@example.co.nz:/$ export -d Pictures/
Disabled export: /Pictures
eg.email_1@example.co.nz:/$ export
Couldn't find anything exported below current folder. Use -a to export it
eg.email_1@example.co.nz:/$
```