smbclient SMB share client

START HERE

`smbclient -L <host> -U <user>`
List shares

`smbclient //<host>/<share> -U <user>`
Open a share

`ls`
List remote files

`cd <dir>`
Change remote dir

`get <file>`
Download a file

`put <file>`
Upload a file

`mget <mask>`
Download many

`exit`
Leave the session

smbclient -L find shares → `pick share` → **//host/share** authenticate → `smb: \>` → **Prompt** ls, get, put

CONCEPTS

Remote commands act on the share; lcd and ! act on your local machine. Quote paths with spaces.

mget and mput ask per file and skip subdirectories until you toggle prompt and recurse.

CONNECT

`smbclient -L <host>`
List shares on a server

`smbclient -L <host> -N`
List shares anonymously
-N skips the password prompt; works only if guest access is allowed.

`smbclient //<host>/<share>`
Open a share interactively

`smbclient //<host>/<share> -U <user>`
Connect as a user
You are prompted for the password.

`smbclient //<host>/<share> -U <domain>\\<user>`
Connect as a domain user
Or pass the domain with -W <workgroup>.

`smbclient //<host>/<share> -N`
Connect without a password

`smbclient //<host>/<share> -A <file>`
Read credentials from a file
File holds username=, password= and domain= lines; keep it owner-readable.

`smbclient //<host>/<share> -I <ip>`
Connect by IP address
Skips name resolution.

`smbclient //<host>/<share> -p <port>`
Use another TCP port

`smbclient //<host>/<share> -m SMB3`
Cap the protocol version
-m sets the highest protocol; SMB2 and NT1 pick older ones.

`smbclient //<host>/<share> --use-kerberos=required`
Authenticate with Kerberos
Use DNS names, not IP addresses.

CONNECT (cont.)

`smbclient //<host>/<share> --client-protection=encrypt`
Require encryption

`smbclient -L <host> -g`
Machine-readable share list

NAVIGATE

`ls [<mask>]`
List remote files

`dir [<mask>]`
Same as ls

`cd <dir>`
Change remote directory
Without an argument it shows the current directory.

`lcd <dir>`
Change local directory
Sets where get and put read and write locally.

`du <dir>`
List with disk usage and free space

`! <command>`
Run a local shell command

`more <file>`
Page a remote file

`help [<command>]`
Show command help

`exit`
End the session

`history`
Show command history

TRANSFER

`get <remote> [<local>]`
Download a file

`put <local> [<remote>]`
Upload a file

`mget <mask>`
Download matching files

`mput <mask>`
Upload matching files

`prompt`
Toggle per-file confirmation
Turn it off before mget or mput to avoid a question per file.

`recurse`
Toggle recursion into subdirectories
Needed for mget and mput to descend into directories.

TRANSFER (cont.)

`mask <mask>`
Filter for recursive transfers

`lowercase`
Lowercase local names on get

`iosize <bytes>`
Set the I/O buffer size

`scopy <src> <dst>`
Copy on the server

MANAGE FILES

`mkdir <dir>`
Create a remote directory

`rmdir <dir>`
Remove a remote directory

`del <mask>`
Delete files

`deltree <mask>`
Delete recursively
Stops at the first failure.

`rename <old> <new>`
Rename or move a file

`allinfo <file>`
Show all file information

`setmode <file> +r`
Set DOS attributes (rsha)

`notify <dir>`
Watch a directory for changes

SCRIPTING

`smbclient //<host>/<share> -c "<cmds>"`
Run commands, then exit
Separate commands with semicolons. Implies -N, so combine with -A or --password.

`smbclient //<host>/<share> -U <user>%<pass>`
Give the password inline
Visible in the process list and shell history; prefer -A.

`smbclient //<host>/<share> -c "ls" -E`
Send messages to stderr

`smbclient //<host>/<share> -d <level>`
Set debug level 0-10

`smbclient //<host>/<share> -t <seconds>`
Per-request timeout

`smbclient //<host>/<share> --option="<name>=<value>"`
Override an smb.conf option

TAR BACKUP

`smbclient //<host>/<share> -Tc <file>.tar`
Back up the share to a tar
Use - as the file name for stdout together with -d0.

`smbclient //<host>/<share> -Tx <file>.tar`
Restore a tar to the share

`smbclient //<host>/<share> -Tc <file>.tar <path>`
Back up only a path

`smbclient //<host>/<share> -Tcg <file>.tar`
Only files with archive bit set

`smbclient //<host>/<share> -Tcn <file>.tar`
Dry run: archive nothing

`tar c <file>.tar`
Tar from inside a session

RECIPES

Download a whole directory

```
smbclient //<host>/<share> -U
<user>
smb: \> cd <dir>
smb: \> recurse ON
smb: \> prompt OFF
smb: \> mget *
```

One-shot upload

```
smbclient //<host>/<share> -A
~/.smbcred \
  -c "cd <dir>; put report.pdf"
```

Back up a share to a local archive

```
smbclient //<host>/<share> -A
~/.smbcred \
  -Tc backup.tar
```

Find shares, then mount one

```
smbclient -L <host> -U <user>
sudo mount -t cifs //<host>/
<share> /mnt \
  -o username=<user>
```

HELP

`smbclient --help`
Option summary

`man smbclient`
Full manual