FTP

FTP is the File Transfer Protocol. Rclone FTP support is provided using the github.com/jlaffaye/ftp package.

Limitations of Rclone's FTP backend

Paths are specified as `remote:path`. If the path does not begin with a `/` it is relative to the home directory of the user. An empty path`remote:` refers to the user's home directory.

To create an FTP configuration named `remote`, run

```
rclone config
```

Rclone config guides you through an interactive setup process. A minimal rclone FTP remote definition only requires host, username and password. For an anonymous FTP server, use `anonymous` as username and your email address as password.

```
No remotes found - make a new one
n) New remote
r) Rename remote
c) Copy remote
s) Set configuration password
q) Quit config
n/r/c/s/q> n
name> remote
Type of storage to configure.
Enter a string value. Press Enter for the default ("").
Choose a number from below, or type in your own value
[snip]
XX / FTP Connection
   \ "ftp"
[snip]
Storage> ftp
** See help for ftp backend at: https://rclone.org/ftp/ **

FTP host to connect to
Enter a string value. Press Enter for the default ("").
Choose a number from below, or type in your own value
 1 / Connect to ftp.example.com
   \ "ftp.example.com"
host> ftp.example.com
FTP username, leave blank for current username, $USER
Enter a string value. Press Enter for the default ("").
user>
FTP port, leave blank to use default (21)
Enter a string value. Press Enter for the default ("").
port>
FTP password
y) Yes type in my own password
g) Generate random password
y/g> y
Enter the password:
password:
Confirm the password:
password:
Use FTP over TLS (Implicit)
Enter a boolean value (true or false). Press Enter for the default ("false").
tls>
Use FTP over TLS (Explicit)
Enter a boolean value (true or false). Press Enter for the default ("false").
explicit_tls>
Remote config
--------------------
[remote]
type = ftp
host = ftp.example.com
pass = *** ENCRYPTED ***
--------------------
```

```
y) Yes this is OK
e) Edit this remote
d) Delete this remote
y/e/d> y
```

To see all directories in the home directory of `remote`

```
rclone lsd remote:
```

Make a new directory

```
rclone mkdir remote:path/to/directory
```

List the contents of a directory

```
rclone ls remote:path/to/directory
```

Sync `/home/local/directory` to the remote directory, deleting any excess files in the directory.

```
rclone sync -i /home/local/directory remote:directory
```

Example without a config file

```
rclone lsf :ftp: --ftp-host=speedtest.tele2.net --ftp-user=anonymous --ftp-pass=`rclone obscu
```

Implicit TLS

Rlone FTP supports implicit FTP over TLS servers (FTPS). This has to be enabled in the FTP backend config for the remote, or with `[--ftp-tls]{#ftp-tls}`. The default FTPS port is `990`, not `21` and can be set with `[--ftp-port]{#ftp-port}`.

Standard Options

Here are the standard options specific to ftp (FTP Connection).

--ftp-host

FTP host to connect to

- Config: host
- Env Var: RCLONE_FTP_HOST
- Type: string
- Default: ""
- Examples:

    - "ftp.example.com"
        - Connect to ftp.example.com

--ftp-user

FTP username, leave blank for current username, $USER

- Config: user
- Env Var: RCLONE_FTP_USER
- Type: string
- Default: ""

--ftp-port

FTP port, leave blank to use default (21)

- Config: port
- Env Var: RCLONE_FTP_PORT
- Type: string
- Default: ""

--ftp-pass

FTP password

NB Input to this must be obscured - see rclone obscure.

- Config: pass
- Env Var: RCLONE_FTP_PASS
- Type: string
- Default: ""

--ftp-tls

Use Implicit FTPS (FTP over TLS) When using implicit FTP over TLS the client connects using TLS right from the start which breaks compatibility with non-TLS-aware servers. This is usually served over port 990 rather than port 21. Cannot be used in combination with explicit FTP.

- Config: tls
- Env Var: RCLONE_FTP_TLS
- Type: bool
- Default: false

--ftp-explicit-tls

Use Explicit FTPS (FTP over TLS) When using explicit FTP over TLS the client explicitly requests security from the server in order to upgrade a plain text connection to an encrypted one. Cannot be used in combination with implicit FTP.

- Config: explicit_tls
- Env Var: RCLONE_FTP_EXPLICIT_TLS
- Type: bool
- Default: false

Advanced Options

Here are the advanced options specific to ftp (FTP Connection).

--ftp-concurrency

Maximum number of FTP simultaneous connections, 0 for unlimited

- Config: concurrency
- Env Var: RCLONE_FTP_CONCURRENCY
- Type: int
- Default: 0

--ftp-no-check-certificate

Do not verify the TLS certificate of the server

- Config: no_check_certificate
- Env Var: RCLONE_FTP_NO_CHECK_CERTIFICATE
- Type: bool
- Default: false

--ftp-disable-epsv

Disable using EPSV even if server advertises support

- Config: disable_epsv
- Env Var: RCLONE_FTP_DISABLE_EPSV
- Type: bool
- Default: false

--ftp-disable-mlsd

Disable using MLSD even if server advertises support

- Config: disable_mlsd
- Env Var: RCLONE_FTP_DISABLE_MLSD
- Type: bool
- Default: false

--ftp-encoding

This sets the encoding for the backend.

See: the encoding section in the overview for more info.

- Config: encoding
- Env Var: RCLONE_FTP_ENCODING
- Type: MultiEncoder
- Default: Slash,Del,Ctl,RightSpace,Dot

Limitations

Modified times are not supported. Times you see on the FTP server through rclone are those of upload.

Rclone's FTP backend does not support any checksums but can compare file sizes.

`rclone about` is not supported by the FTP backend. Backends without this capability cannot determine free space for an rclone mount or use policy `mfs` (most free space) as a member of an rclone union remote.

See List of backends that do not support rclone about See rclone about

The implementation of : `--dump headers`, `--dump bodies`, `--dump auth` for debugging isn't the same as for rclone HTTP based backends - it has less fine grained control.

`--timeout` isn't supported (but `--contimeout` is).

`--bind` isn't supported.

Rclone's FTP backend could support server-side move but does not at present.

The `ftp_proxy` environment variable is not currently supported.

FTP servers acting as rclone remotes must support 'passive' mode. Rclone's FTP implementation is not compatible with 'active' mode.

Restricted filename characters

In addition to the default restricted characters set the following characters are also replaced:

File names cannot end with the following characters. Repacement is limited to the last character in a file name:

Character	Value	Replacement
SP	0x20	␠

Not all FTP servers can have all characters in file names, for example:

FTP Server	Forbidden characters
proftpd	`*`
pureftpd	`\ [ ]`