

NAME

fetch - retrieve a file by Uniform Resource Locator

SYNOPSIS

```
fetch [-146AadFIMmnPpqRrsUv] [-B bytes] [--bind-address=host] [--ca-cert=file] [--ca-path=dir]
  [--cert=file] [--crl=file] [-i file] [--key=file] [-N file] [--no-passive] [--no-proxy=list] [--no-ssl3]
  [--no-tls1] [--no-verify-hostname] [--no-verify-peer] [-o file] [--referer=URL] [-S bytes]
  [-T seconds] [--user-agent=agent-string] [-w seconds] URL ...
fetch [-146AadFIMmnPpqRrsUv] [-B bytes] [--bind-address=host] [--ca-cert=file] [--ca-path=dir]
  [--cert=file] [--crl=file] [-i file] [--key=file] [-N file] [--no-passive] [--no-proxy=list] [--no-ssl3]
  [--no-tls1] [--no-verify-hostname] [--no-verify-peer] [-o file] [--referer=URL] [-S bytes]
  [-T seconds] [--user-agent=agent-string] [-w seconds] -h host -f file [-c dir]
```

DESCRIPTION

The **fetch** utility provides a command-line interface to the fetch(3) library. Its purpose is to retrieve the file(s) pointed to by the URL(s) on the command line.

The following options are available:

-1, --one-file

Stop and return exit code 0 at the first successfully retrieved file.

-4, --ipv4-only

Forces **fetch** to use IPv4 addresses only.

-6, --ipv6-only

Forces **fetch** to use IPv6 addresses only.

-A, --no-redirect

Do not automatically follow “temporary” (302) redirects. Some broken Web sites will return a redirect instead of a not-found error when the requested object does not exist.

-a, --retry Automatically retry the transfer upon soft failures.

-B bytes, --buffer-size=bytes

Specify the read buffer size in bytes. The default is 16,384 bytes. Attempts to set a buffer size lower than this will be silently ignored. The number of reads actually performed is reported at verbosity level two or higher (see the **-v** flag).

--bind-address=host

Specifies a hostname or IP address to which sockets used for outgoing connections will be bound.

-c *dir* The file to retrieve is in directory *dir* on the remote host. This option is deprecated and is provided for backward compatibility only.

--ca-cert=*file*

[SSL] Path to certificate bundle containing trusted CA certificates. If not specified, */usr/local/etc/ssl/cert.pem* is used. If this file does not exist, */etc/ssl/cert.pem* is used instead. If neither file exists and no CA path has been configured, OpenSSL's default CA cert and path settings apply. The certificate bundle can contain multiple CA certificates. The *security/ca_root_nss* port is a common source of a current CA bundle.

--ca-path=*dir*

[SSL] The directory *dir* contains trusted CA hashes.

--cert=*file* [SSL] *file* is a PEM encoded client certificate/key which will be used in client certificate authentication.

--crl=*file* [SSL] Points to certificate revocation list *file*, which has to be in PEM format and may contain peer certificates that have been revoked.

-d, --direct Use a direct connection even if a proxy is configured.

-F, --force-restart

In combination with the **-r** flag, forces a restart even if the local and remote files have different modification times. Implies **-R**.

-f *file* The file to retrieve is named *file* on the remote host. This option is deprecated and is provided for backward compatibility only.

-h *host* The file to retrieve is located on the host *host*. This option is deprecated and is provided for backward compatibility only.

-i *file*, **--if-modified-since=***file*

If-Modified-Since mode: the remote file will only be retrieved if it is newer than *file* on the local host. (HTTP only)

--key=*file* [SSL] *file* is a PEM encoded client key that will be used in client certificate authentication in case key and client certificate are stored separately.

-l, --symlink If the target is a file-scheme URL, make a symbolic link to the target rather than trying to copy it.

-M

-m, --mirror Mirror mode: if the file already exists locally and has the same size and modification time as the remote file, it will not be fetched. Note that the **-m** and **-r** flags are mutually exclusive.

-N file, --netrc=file

Use *file* instead of *~/.netrc* to look up login names and passwords for FTP sites. See `ftp(1)` for a description of the file format. This feature is experimental.

-n, --no-mtime

Do not preserve the modification time of the transferred file.

--no-passive Forces the FTP code to use active mode.

--no-proxy=list

Either a single asterisk, which disables the use of proxies altogether, or a comma- or whitespace-separated list of hosts for which proxies should not be used.

--no-sslv3 [SSL] Do not allow SSL version 3 when negotiating the connection. This option is deprecated and is provided for backward compatibility only. SSLv3 is disabled by default. Set `SSL_ALLOW_SSL3` to change this behavior.

--no-tlsv1 [SSL] Do not allow TLS version 1 when negotiating the connection.

--no-verify-hostname

[SSL] Do not verify that the hostname matches the subject of the certificate presented by the server.

--no-verify-peer

[SSL] Do not verify the peer certificate against trusted CAs.

-o file, --output=file

Set the output file name to *file*. By default, a “pathname” is extracted from the specified URI, and its basename is used as the name of the output file. A *file* argument of ‘-’ indicates that results are to be directed to the standard output. If the *file* argument is a directory, fetched file(s) will be placed within the directory, with name(s) selected as in the

default behaviour.

-P

-p, --passive Use passive FTP. These flags have no effect, since passive FTP is the default, but are provided for compatibility with earlier versions where active FTP was the default. To force active mode, use the **--no-passive** flag or set the `FTP_PASSIVE_MODE` environment variable to 'NO'.

--referer=URL

Specifies the referrer URL to use for HTTP requests. If *URL* is set to "auto", the document URL will be used as referrer URL.

-q, --quiet Quiet mode.

-R, --keep-output

The output files are precious, and should not be deleted under any circumstances, even if the transfer failed or was incomplete.

-r, --restart Restart a previously interrupted transfer. Note that the **-m** and **-r** flags are mutually exclusive.

-S bytes, --require-size=bytes

Require the file size reported by the server to match the specified value. If it does not, a message is printed and the file is not fetched. If the server does not support reporting file sizes, this option is ignored and the file is fetched unconditionally.

-s, --print-size

Print the size in bytes of each requested file, without fetching it.

-T seconds, --timeout=seconds

Set timeout value to *seconds*. Overrides the environment variables `FTP_TIMEOUT` for FTP transfers or `HTTP_TIMEOUT` for HTTP transfers if set.

-U, --passive-portrange-default

When using passive FTP, allocate the port for the data connection from the low (default) port range. See `ip(4)` for details on how to specify which port range this corresponds to.

--user-agent=agent-string

Specifies the User-Agent string to use for HTTP requests. This can be useful when

working with HTTP origin or proxy servers that differentiate between user agents.

-v, --verbose

Increase verbosity level.

-w *seconds*, --retry-delay=*seconds*

When the **-a** flag is specified, wait this many seconds between successive retries.

If **fetch** receives a SIGINFO signal (see the **status** argument for `stty(1)`), the current transfer rate statistics will be written to the standard error output, in the same format as the standard completion message.

ENVIRONMENT

FTP_TIMEOUT Maximum time, in seconds, to wait before aborting an FTP connection.

HTTP_TIMEOUT Maximum time, in seconds, to wait before aborting an HTTP connection.

See `fetch(3)` for a description of additional environment variables, including `FETCH_BIND_ADDRESS`, `FTP_LOGIN`, `FTP_PASSIVE_MODE`, `FTP_PASSWORD`, `FTP_PROXY`, `ftp_proxy`, `HTTP_ACCEPT`, `HTTP_AUTH`, `HTTP_PROXY`, `http_proxy`, `HTTP_PROXY_AUTH`, `HTTP_REFERER`, `HTTP_USER_AGENT`, `NETRC`, `NO_PROXY`, `no_proxy`, `SSL_CA_CERT_FILE`, `SSL_CA_CERT_PATH`, `SSL_CLIENT_CERT_FILE`, `SSL_CLIENT_KEY_FILE`, `SSL_CRL_FILE`, `SSL_ALLOW_SSL3`, `SSL_NO_TLS1`, `SSL_NO_TLS1_1`, `SSL_NO_TLS1_2`, `SSL_NO_VERIFY_HOSTNAME` and `SSL_NO_VERIFY_PEER`.

EXIT STATUS

The **fetch** command returns zero on success, or one on failure. If multiple URLs are listed on the command line, **fetch** will attempt to retrieve each one of them in turn, and will return zero only if they were all successfully retrieved.

If the **-i** argument is used and the remote file is not newer than the specified file then the command will still return success, although no file is transferred.

EXAMPLES

Silently try to fetch the URLs passed as parameters. The first one will fail. If the second URL succeeds the third one will not be tried:

```
$ fetch -l -q https://www.freebsd.org/bad.html \  
    ftp.freebsd.org/pub/FreeBSD/README.TXT \  
    https://www.fake.url  
fetch: https://www.freebsd.org/bad.html: Not Found
```

Be verbose when retrieving the *README.TXT* file:

```
$ fetch -v ftp.freebsd.org/pub/FreeBSD/README.TXT
resolving server address: ftp.freebsd.org:80
requesting http://ftp.freebsd.org/pub/FreeBSD/README.TXT
local size / mtime: 4259 / 1431015519
remote size / mtime: 4259 / 1431015519
README.TXT                4259 B  44 MBps  00s
```

Quietly save the *README.TXT* file as *myreadme.txt* and do not delete the output file under any circumstances:

```
fetch -o myreadme.txt -q -R ftp.freebsd.org/pub/FreeBSD/README.TXT
```

Print the size of the requested file and identify the request with a custom user agent string:

```
$ fetch -s ftp.freebsd.org/pub/FreeBSD/README.TXT
--user-agent="Mozilla/5.0 (X11; FreeBSD x86_64; rv:78.0) Gecko/20100101"
3513231
```

Restart the transfer of the *README.TXT* file and retry the transfer upon soft failures:

```
$ fetch -a -r http://ftp.freebsd.org/pub/FreeBSD/README.TXT
```

SEE ALSO

fetch(3), phhttpget(8)

HISTORY

The **fetch** command appeared in FreeBSD 2.1.5. This implementation first appeared in FreeBSD 4.1.

AUTHORS

The original implementation of **fetch** was done by Jean-Marc Zucconi <jmz@FreeBSD.org>. It was extensively re-worked for FreeBSD 2.2 by Garrett Wollman <wollman@FreeBSD.org>, and later completely rewritten to use the fetch(3) library by Dag-Erling Smørgrav <des@FreeBSD.org> and Michael Gmelin <freebsd@grem.de>.

NOTES

The **-b** and **-t** options are no longer supported and will generate warnings. They were workarounds for bugs in other OSes which this implementation does not trigger.

One cannot both use the **-h**, **-c** and **-f** options and specify URLs on the command line.