



Linux Ubuntu 26.04 Manual Pages on command 'git-upload-pack.1'

\$ man git-upload-pack.1

GIT-UPLOAD-PACK(1)

Git Manual

GIT-UPLOAD-PACK(1)

NAME

git-upload-pack - Send objects packed back to git-fetch-pack

SYNOPSIS

```
git-upload-pack [--no-strict] [--timeout=<n>] [--stateless-rpc]
                [--advertise-refs] <directory>
```

DESCRIPTION

Invoked by git fetch-pack, learns what objects the other side is missing, and sends them after packing.

This command is usually not invoked directly by the end user. The UI for the protocol is on the git fetch-pack side, and the program pair is meant to be used to pull updates from a remote repository. For push operations, see git send-pack.

OPTIONS

--strict, --no-strict

Do not try <directory>/.git/ if <directory> is not a Git directory.

--timeout=<n>

Interrupt transfer after <n> seconds of inactivity.

--stateless-rpc

Perform only a single read-write cycle with stdin and stdout. This fits with the HTTP

POST request processing model where a program may read the request, write a response, and must exit.

--http-backend-info-refs

Used by git-http-backend(1) to serve up \$GIT_URL/info/refs?service=git-upload-pack

requests. See "Smart Clients" in gitprotocol-http(5) and "HTTP Transport" in the gitprotocol-v2(5) documentation. Also understood by git-receive-pack(1).

<directory>

The repository to sync from.

ENVIRONMENT

GIT_PROTOCOL

Internal variable used for handshaking the wire protocol. Server admins may need to configure some transports to allow this variable to be passed. See the discussion in git(1).

GIT_NO_LAZY_FETCH

When cloning or fetching from a partial repository (i.e., one itself cloned with --filter), the server-side upload-pack may need to fetch extra objects from its upstream in order to complete the request. By default, upload-pack will refuse to perform such a lazy fetch, because git fetch may run arbitrary commands specified in configuration and hooks of the source repository (and upload-pack tries to be safe to run even in untrusted .git directories).

This is implemented by having upload-pack internally set the GIT_NO_LAZY_FETCH variable to 1. If you want to override it (because you are fetching from a partial clone, and you are sure you trust it), you can explicitly set GIT_NO_LAZY_FETCH to 0.

SECURITY

Most Git commands should not be run in an untrusted .git directory (see the section SECURITY in git(1)). upload-pack tries to avoid any dangerous configuration options or hooks from the repository it's serving, making it safe to clone an untrusted directory and run commands on the resulting clone.

For an extra level of safety, you may be able to run upload-pack as an alternate user. The details will be platform dependent, but on many systems you can run:

```
git clone --no-local --upload-pack='sudo -u nobody git-upload-pack' ...
```

SEE ALSO

gitnamespaces(7)

GIT

Part of the git(1) suite