Rocky Enterprise Linux 9.2 Manual Pages on command 'git-fetch-pack.1'

C:\>man git-fetch-pack.1

GIT-FETCH-PACK(1) Git Manual GIT-FETCH-PACK(1)

NAME

git-fetch-pack - Receive missing objects from another repository

SYNOPSIS

git fetch-pack [--all] [--quiet|-q] [--keep|-k] [--thin] [--include-tag]

[--upload-pack=<git-upload-pack>]

[--depth=<n>] [--no-progress]

[-v] <repository> [<refs>...]

DESCRIPTION

Usually you would want to use git fetch, which is a higher level wrapper of this command, instead.

Invokes git-upload-pack on a possibly remote repository and asks it to send objects missing from this repository, to update the named heads. The list of commits available locally is found out by scanning the local refs/ hierarchy and sent to git-upload-pack running on the other end.

This command degenerates to download everything to complete the asked refs from the remote side when the local side does not have a common ancestor commit.

OPTIONS

--all

Fetch all remote refs.

--stdin

Take the list of refs from stdin, one per line. If there are refs specified on

the command line in addition to this option, then the refs from stdin are processed after those on the command line.

If --stateless-rpc is specified together with this option then the list of refs must be in packet format (pkt-line). Each ref must be in a separate packet, and the list must end with a flush packet.

-q, --quiet

Pass -q flag to git unpack-objects; this makes the cloning process less verbose.

-k, --keep

Do not invoke git unpack-objects on received data, but create a single packfile out of it instead, and store it in the object database. If provided twice then the pack is locked against repacking.

--thin

Fetch a "thin" pack, which records objects in deltified form based on objects not included in the pack to reduce network traffic.

--include-tag

If the remote side supports it, annotated tags objects will be downloaded on the same connection as the other objects if the object the tag references is downloaded. The caller must otherwise determine the tags this option made available.

--upload-pack=<git-upload-pack>

Use this to specify the path to git-upload-pack on the remote side, if is not found on your $PATH. Installations of sshd ignores the user?s environment setup scripts for login shells (e.g. .bash_profile) and your privately installed git may not be found on the system default $PATH. Another workaround suggested is to set up your $PATH in ".bashrc", but this flag is for people who do not want to pay the overhead for non-interactive shells by having a lean .bashrc file (they set most of the things up in .bash_profile).

--exec=<git-upload-pack>

Same as --upload-pack=<git-upload-pack>.

--depth=<n>

Limit fetching to ancestor-chains not longer than n. git-upload-pack treats the special depth 2147483647 as infinite even if there is an ancestor-chain

that long.

--shallow-since=<date>

Deepen or shorten the history of a shallow repository to include all reachable commits after <date>.

--shallow-exclude=<revision>

Deepen or shorten the history of a shallow repository to exclude commits reachable from a specified remote branch or tag. This option can be specified multiple times.

--deepen-relative

Argument --depth specifies the number of commits from the current shallow boundary instead of from the tip of each remote branch history.

--no-progress

Do not show the progress.

--check-self-contained-and-connected

Output "connectivity-ok" if the received pack is self-contained and connected.

-v

Run verbosely.

<repository>

The URL to the remote repository.

<refs>...

The remote heads to update from. This is relative to $GIT_DIR (e.g. "HEAD", "refs/heads/master"). When unspecified, update from all heads the remote side has.

If the remote has enabled the options uploadpack.allowTipSHA1InWant, uploadpack.allowReachableSHA1InWant, or uploadpack.allowAnySHA1InWant, they may alternatively be 40-hex sha1s present on the remote.

SEE ALSO

git-fetch(1)

GIT

Part of the git(1) suite