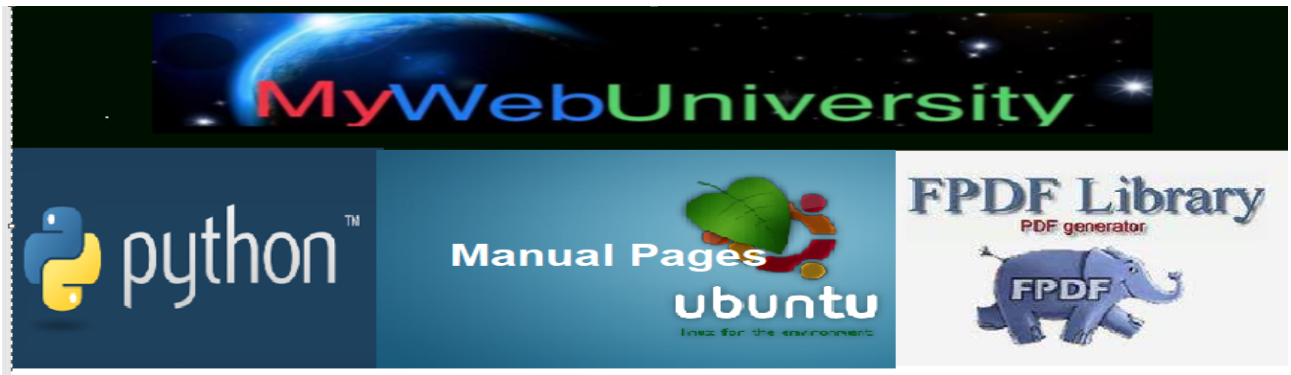

Full credit is given to the above companies including the OS that this PDF file was generated!

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

$ man git-pack-refs.1

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

NAME

git-pack-refs - Pack heads and tags for efficient repository access

SYNOPSIS

git pack-refs [--all] [--no-prune] [--auto] [--include <pattern>] [--exclude <pattern>]

DESCRIPTION

Traditionally, tips of branches and tags (collectively known as refs) were stored one file per ref in a (sub)directory under $GIT_DIR/refs directory. While many branch tips tend to be updated often, most tags and some branch tips are never updated. When a repository has hundreds or thousands of tags, this one-file-per-ref format both wastes storage and hurts performance.

This command is used to solve the storage and performance problem by storing the refs in a single file, $GIT_DIR/packed-refs. When a ref is missing from the traditional $GIT_DIR/refs directory hierarchy, it is looked up in this file and used if found.

Subsequent updates to branches always create new files under $GIT_DIR/refs directory hierarchy.

A recommended practice to deal with a repository with too many refs is to pack its refs with --all once, and occasionally run git pack-refs. Tags are by definition stationary and are not expected to change. Branch heads will be packed with the initial pack-refs --all, but only the currently active branch heads will become unpacked, and the next pack-refs (without --all) will leave them unpacked.

OPTIONS

--all

The command by default packs all tags and refs that are already packed, and leaves other refs alone. This is because branches are expected to be actively developed and packing their tips does not help performance. This option causes all refs to be packed as well, with the exception of hidden refs, broken refs, and symbolic refs. Useful for a repository with many branches of historical interests.

--no-prune

The command usually removes loose refs under $GIT_DIR/refs hierarchy after packing them. This option tells it not to.

--auto

Pack refs as needed depending on the current state of the ref database. The behavior depends on the ref format used by the repository and may change in the future.

? "files": Loose references are packed into the packed-refs file based on the ratio of loose references to the size of the packed-refs file. The bigger the packed-refs file, the more loose references need to exist before we repack.

? "reftable": Tables are compacted such that they form a geometric sequence. For two tables N and N+1, where N+1 is newer, this maintains the property that N is at least twice as big as N+1. Only tables that violate this property are compacted.

--include <pattern>

Pack refs based on a glob(7) pattern. Repetitions of this option accumulate inclusion patterns. If a ref is both included in --include and --exclude, --exclude takes precedence. Using --include will preclude all tags from being included by default. Symbolic refs and broken refs will never be packed. When used with --all, it will be a noop. Use --no-include to clear and reset the list of patterns.

--exclude <pattern>

Do not pack refs matching the given glob(7) pattern. Repetitions of this option accumulate exclusion patterns. Use --no-exclude to clear and reset the list of patterns. If a ref is already packed, including it with --exclude will not unpack it.

When used with --all, pack only loose refs which do not match any of the provided --exclude patterns.

When used with --include, refs provided to --include, minus refs that are provided to --exclude will be packed.

BUGS

Older documentation written before the packed-refs mechanism was introduced may still say

things like ".git/refs/heads/<branch> file exists" when it means "branch <branch> exists".

GIT

Part of the git(1) suite