

NAME

git-repack - Pack unpacked objects in a repository

SYNOPSIS

```
git repack [-a] [-A] [-d] [-f] [-F] [-l] [-n] [-q] [-b] [--window=<n>] [--depth=<n>]
```

DESCRIPTION

This command is used to combine all objects that do not currently reside in a pack, into a pack. It can also be used to re-organize existing packs into a single, more efficient pack.

A pack is a collection of objects, individually compressed, with delta compression applied, stored in a single file, with an associated index file.

Packs are used to reduce the load on mirror systems, backup engines, disk storage, etc.

OPTIONS

-a

Instead of incrementally packing the unpacked objects, pack everything referenced into a single pack. Especially useful when packing a repository that is used for private development. Use with *-d*. This will clean up the objects that git prune leaves behind, but git fsck --full --dangling shows as dangling.

Note that users fetching over dumb protocols will have to fetch the whole new pack in order to get any contained object, no matter how many other objects in that pack they already have locally.

-A

Same as *-a*, unless *-d* is used. Then any unreachable objects in a previous pack become loose, unpacked objects, instead of being left in the old pack. Unreachable objects are never intentionally added to a pack, even when repacking. This option prevents unreachable objects from being immediately deleted by way of being left in the old pack and then removed. Instead, the loose unreachable objects will be pruned according to normal expiry rules with the next *git gc* invocation. See [git-gc\(1\)](#).

-d

After packing, if the newly created packs make some existing packs redundant, remove the redundant packs. Also run *git prune-packed* to remove redundant loose object files.

-l

Pass the --local option to *git pack-objects*. See [git-pack-objects\(1\)](#).

-f

Pass the --no-reuse-delta option to *git-pack-objects*, see [git-pack-objects\(1\)](#).

-F

Pass the --no-reuse-object option to *git-pack-objects*, see [git-pack-objects\(1\)](#).

-q

Pass the -q option to *git pack-objects*. See [git-pack-objects\(1\)](#).

-n

Do not update the server information with *git update-server-info*. This option skips updating local catalog files needed to publish this repository (or a direct copy of it) over HTTP or FTP. See [git-update-server-info\(1\)](#).

--window=<n>, --depth=<n>

These two options affect how the objects contained in the pack are stored using delta compression. The objects are first internally sorted by type, size and optionally names and compared against the other objects within --window to see if using delta compression saves space. --depth limits the maximum delta depth; making it too deep affects the performance on the unpacker side, because delta data needs to be applied that many times to get to the necessary object. The default value for --window is 10 and --depth is 50.

`--window-memory=<n>`

This option provides an additional limit on top of `--window`; the window size will dynamically scale down so as to not take up more than `<n>` bytes in memory. This is useful in repositories with a mix of large and small objects to not run out of memory with a large window, but still be able to take advantage of the large window for the smaller objects. The size can be suffixed with k, m, or g. `--window-memory=0` makes memory usage unlimited, which is the default.

`--max-pack-size=<n>`

Maximum size of each output pack file. The size can be suffixed with k, m, or g. The minimum size allowed is limited to 1 MiB. If specified, multiple packfiles may be created. The default is unlimited, unless the config variable `pack.packSizeLimit` is set.

`-b, --write-bitmap-index`

Write a reachability bitmap index as part of the repack. This only makes sense when used with `-a` or `-A`, as the bitmaps must be able to refer to all reachable objects. This option overrides the setting of `pack.writebitmaps`.

`--pack-kept-objects`

Include objects in `.keep` files when repacking. Note that we still do not delete `.keep` packs after `pack-objects` finishes. This means that we may duplicate objects, but this makes the option safe to use when there are concurrent pushes or fetches. This option is generally only useful if you are writing bitmaps with `-b` or `pack.writebitmaps`, as it ensures that the bitmapped packfile has the necessary objects.

CONFIGURATION

By default, the command passes `--delta-base-offset` option to *git pack-objects*; this typically results in slightly smaller packs, but the generated packs are incompatible with versions of Git older than version 1.4.4. If you need to share your repository with such ancient Git versions, either directly or via the dumb http or rsync protocol, then you need to set the configuration variable `repack.UseDeltaBaseOffset` to false and `repack`. Access from old Git versions over the native protocol is unaffected by this option as the conversion is performed on the fly as needed in that case.

SEE ALSO

[git-pack-objects\(1\)](#) [git-prune-packed\(1\)](#)

GIT

Part of the [git\(1\)](#) suite