git-checkout-index

git-checkout-index(1) ==========

NAME ---- git-checkout-index - Copy files from the index to the working tree

SYNOPSIS --------

> 'git checkout-index' [-u] [-q] [-a] [-f] [-n] [--prefix=<string>]
> [--stage=<number>|all]
> [--temp]
> [--ignore-skip-worktree-bits]
> [-z] [--stdin]
> [--] [<file>...]

DESCRIPTION ----------- Copies all listed files from the index to the working directory (not overwriting existing files).

OPTIONS ------- -u:: --index:: update stat information for the checked out entries in the index file.

-q

--quiet
be quiet if files exist or are not in the index

-f

--force
forces overwrite of existing files

-a

--all
checks out all files in the index except for those with the skip-worktree bit set (see `--ignore-skip-worktree-bits`). Cannot be used together with explicit filenames.

-n

--no-create
Don't checkout new files, only refresh files already checked out.

--prefix=<string>
When creating files, prepend <string> (usually a directory including a trailing /)

--stage=<number>|all
Instead of checking out unmerged entries, copy out the files from the named stage. <number> must be between 1 and 3. Note: --stage=all automatically implies --temp.

--temp
Instead of copying the files to the working directory, write the content to temporary files. The temporary name associations will be written to stdout.

--ignore-skip-worktree-bits
Check out all files, including those with the skip-worktree bit set.

--stdin
Instead of taking a list of paths from the command line, read the list of paths from the standard input. Paths are separated by LF (i.e. one path per line) by default.

-z
Only meaningful with `--stdin`; paths are separated with NUL character instead of LF.

--
Do not interpret any more arguments as options.

The order of the flags used to matter, but not anymore.

Just doing `git` `checkout-index` does nothing. You probably meant `git` `checkout-index` `-a`. And if you want to force it, you want `git` `checkout-index` `-f` `-a`.

Intuitiveness is not the goal here. Repeatability is. The reason for the "no arguments means no work" behavior is that from scripts you are supposed to be able to do:

```
$ find . -name '*.h' -print0 | xargs -0 git checkout-index -f --
```

which will force all existing `*.h` files to be replaced with their cached copies. If an empty command line implied "all", then this would force-refresh everything in the index, which was not the point. But since ’git checkout-index’ accepts --stdin it would be faster to use:

```
$ find . -name '*.h' -print0 | git checkout-index -f -z --stdin
```

The `--` is just a good idea when you know the rest will be filenames; it will prevent problems with a filename of, for example, `-a`. Using `--` is probably a good policy in scripts.

Using --temp or --stage=all ---------- When `--temp` is used (or implied by `--stage=all`) ’git checkout-index’ will create a temporary file for each index entry being checked out. The index will not be updated with stat information. These options can be useful if the caller needs all stages

of all unmerged entries so that the unmerged files can be processed by an external merge tool.

A listing will be written to stdout providing the association of temporary file names to tracked path names. The listing format has two variations:

1. tempname TAB path RS
   The first format is what gets used when `--stage` is omitted or is not `--stage=all`. The field tempname is the temporary file name holding the file content and path is the tracked path name in the index. Only the requested entries are output.

1. stage1temp SP stage2temp SP stage3tmp TAB path RS
   The second format is what gets used when `--stage=all`. The three stage temporary fields (stage1temp, stage2temp, stage3temp) list the name of the temporary file if there is a stage entry in the index or `.` if there is no stage entry. Paths which only have a stage 0 entry will always be omitted from the output.

In both formats RS (the record separator) is newline by default but will be the null byte if -z was passed on the command line. The temporary file names are always safe strings; they will never contain directory separators or whitespace characters. The path field is always relative to the current directory and the temporary file names are always relative to the top level directory.

If the object being copied out to a temporary file is a symbolic link the content of the link will be written to a normal file. It is up to the end-user or the Porcelain to make use of this information.

EXAMPLES -------- To update and refresh only the files already checked out::
---------- $ git checkout-index -n -f -a && git update-index --ignore-missing --refresh ----------

Using ’git checkout-index’ to "export an entire tree"
The prefix ability basically makes it trivial to use ’git checkout-index’ as an "export as tree" function. Just read the desired tree into the index, and do:

```
$ git checkout-index --prefix=git-export-dir/ -a
```

`git` `checkout-index` will "export" the index into the specified directory.

The final "/" is important. The exported name is literally just prefixed with the specified string. Contrast this with the following example.

Export files with a prefix
$ git checkout-index --prefix=.merged- Makefile

This will check out the currently cached copy of `Makefile` into the file `.merged-Makefile`.

GIT --- Part of the linkgit:git[1] suite

□□□ #1
□ □□□ □□□ 2026-10-31 20:10:04 UTC
□ □□□ □□□ 2026-10-31 20:10:04 UTC