



Linux Ubuntu 26.04 Manual Pages on command 'npm-diff.1'

\$ man npm-diff.1

NPM-DIFF(1)

General Commands Manual

NPM-DIFF(1)

NAME

npm-diff

Synopsis

```
<!-- AUTOGENERATED USAGE DESCRIPTIONS -->
```

Description

Similar to its git diff counterpart, this command will print diff patches of files for packages published to the npm registry.

```
? npm diff --diff=<spec-a> --diff=<spec-b>
```

Compares two package versions using their registry specifiers, e.g:

npm diff --diff=pkg@1.0.0 --diff=pkg@^2.0.0. It's also possible to compare across forks of any package,

e.g: npm diff --diff=pkg@1.0.0 --diff=pkg-fork@1.0.0.

Any valid spec can be used, so that it's also possible to compare directories or git repositories,

e.g: npm diff --diff=pkg@latest --diff=./packages/pkg

Here's an example comparing two different versions of a package named abbrev from the registry:

```
npm diff --diff=abbrev@1.1.0 --diff=abbrev@1.1.1
```

On success, output looks like:

```
diff --git a/package.json b/package.json
```

```
index v1.1.0..v1.1.1 100644
```

```
--- a/package.json
```

```

+++ b/package.json
@@ -1,6 +1,6 @@
{
  "name": "abbrev",
-  "version": "1.1.0",
+  "version": "1.1.1",
  "description": "Like ruby's abbrev module, but in js",
  "author": "Isaac Z. Schlueter <i@izs.me>",
  "main": "abbrev.js",

```

Given the flexible nature of npm specs, you can also target local directories or git repos just like when using `npm install`:

```
npm diff --diff=https://github.com/npm/libnpmdiff --diff=./local-path
```

In the example above we can compare the contents from the package installed from the git repo at `github.com/npm/libnpmdiff` with the contents of the `./local-path` that contains a valid package, such as a modified copy of the original.

? `npm diff` (in a package directory, no arguments):

If the package is published to the registry, `npm diff` will fetch the tarball version tagged as latest (this value can be configured using the `tag` option) and proceed to compare the contents of files present in that tarball, with the current files in your local file system.

This workflow provides a handy way for package authors to see what package-tracked files have been changed in comparison with the latest published version of that package.

? `npm diff --diff=<pkg-name>` (in a package directory):

When using a single package name (with no version or tag specifier) as an argument, `npm diff` will work in a similar way to

``npm-outdated`` and reach for the registry to figure out what current published version of the package named `<pkg-name>` will satisfy its dependent declared semver-range. Once that specific version is known `npm diff` will print diff patches comparing the current version of `<pkg-name>` found in the local file system with that specific version returned by the registry.

Given a package named abbrev that is currently installed:

```
npm diff --diff=abbrev
```

That will request from the registry its most up to date version and will print a diff output comparing the currently installed version to this newer one if the version numbers are not the same.

? npm diff --diff=<spec-a> (in a package directory):

Similar to using only a single package name, it's also possible to declare a full registry specifier version if you wish to compare the local version of an installed package with the specific version/tag/semver-range provided in <spec-a>.

An example: assuming pkg@1.0.0 is installed in the current node_modules folder, running:

```
npm diff --diff=pkg@2.0.0
```

It will effectively be an alias to

```
npm diff --diff=pkg@1.0.0 --diff=pkg@2.0.0.
```

? npm diff --diff=<semver-a> [--diff=<semver-b>] (in a package directory):

Using npm diff along with semver-valid version numbers is a shorthand to compare different versions of the current package.

It needs to be run from a package directory, such that for a package named pkg running npm diff --diff=1.0.0 --diff=1.0.1 is the same as running npm diff --diff=pkg@1.0.0 --diff=pkg@1.0.1.

If only a single argument <version-a> is provided, then the current local file system is going to be compared against that version.

Here's an example comparing two specific versions (published to the configured registry) of the current project directory:

```
npm diff --diff=1.0.0 --diff=1.1.0
```

Note that tag names are not valid --diff argument values, if you wish to compare to a published tag, you must use the pkg@tagname syntax.

Filtering files

It's possible to also specify positional arguments using file names or globs pattern matching in order to limit the result of diff patches to only a subset of files for a given package, e.g:

```
npm diff --diff=pkg@2 ./lib/ CHANGELOG.md
```

In the example above the diff output is only going to print contents of files located within the folder `./lib/` and changed lines of code within the `CHANGELOG.md` file.

Configuration

<!-- AUTOGENERATED CONFIG DESCRIPTIONS -->

See Also

- ? npm outdated
- ? npm install
- ? npm config
- ? npm registry

9.2.0

March 2026

NPM-DIFF(1)