



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

\$ man npm-ls.1

NPM-LS(1)

General Commands Manual

NPM-LS(1)

NAME

npm-ls

Synopsis

<!-- AUTOGENERATED USAGE DESCRIPTIONS -->

Description

This command will print to stdout all the versions of packages that are installed, as well as their dependencies when --all is specified, in a tree structure.

Note: to get a "bottoms up" view of why a given package is included in the tree at all, use npm explain.

Positional arguments are name@version-range identifiers, which will limit the results to only the paths to the packages named. Note that nested packages will also show the paths to the specified packages. For example, running npm ls promzard in npm's source tree will show:

```
npm@@VERSION@ /path/to/npm
```

```
??? init-package-json@0.0.4
```

```
??? promzard@0.1.5
```

It will print out extraneous, missing, and invalid packages.

If a project specifies git urls for dependencies these are shown in parentheses after the name@version to make it easier for users to recognize potential forks of a project.

The tree shown is the logical dependency tree, based on package

dependencies, not the physical layout of your node_modules folder.

When run as ll or la, it shows extended information by default.

Note: Design Changes Pending

The npm ls command's output and behavior made a ton of sense when npm created a node_modules folder that naively nested every dependency. In such a case, the logical dependency graph and physical tree of packages on disk would be roughly identical.

With the advent of automatic install-time deduplication of dependencies in npm v3, the ls output was modified to display the logical dependency graph as a tree structure, since this was more useful to most users.

However, without using npm ls -l, it became impossible to show where a package was actually installed much of the time!

With the advent of automatic installation of peerDependencies in npm v7, this gets even more curious, as peerDependencies are logically "underneath" their dependents in the dependency graph, but are always physically at or above their location on disk.

Also, in the years since npm got an ls command (in version 0.0.2!), dependency graphs have gotten much larger as a general rule. Therefore, in order to avoid dumping an excessive amount of content to the terminal, npm ls now only shows the top level dependencies, unless --all is provided.

A thorough re-examination of the use cases, intention, behavior, and output of this command, is currently underway. Expect significant changes to at least the default human-readable npm ls output in npm v8.

Configuration

```
<!-- AUTOGENERATED CONFIG DESCRIPTIONS -->
```

See Also

- ? package spec
- ? npm explain
- ? npm config
- ? npmrc
- ? npm folders
- ? npm explain

- ? npm install
- ? npm link
- ? npm prune
- ? npm outdated
- ? npm update

9.2.0

March 2026

NPM-LS(1)