

Description

npm gets its configuration values from the following sources, sorted by priority:

Command Line Flags

Putting `--foo bar` on the command line sets the `foo` configuration parameter to `"bar"`. A `--` argument tells the cli parser to stop reading flags. Using `--flag` without specifying any value will set the value to `true`.

Example: `--flag1 --flag2` will set both configuration parameters to `true`, while `--flag1 --flag2 bar` will set `flag1` to `true`, and `flag2` to `bar`. Finally, `--flag1 --flag2 -- bar` will set both configuration parameters to `true`, and the `bar` is taken as a command argument.

Environment Variables

Any environment variables that start with `npm_config_` will be interpreted as a configuration parameter. For example, putting `npm_config_foo=bar` in your environment will set the `foo` configuration parameter to `bar`. Any environment configurations that are not given a value will be given the value of `true`. Config values are case-insensitive, so `NPM_CONFIG_FOO=bar` will work the same. However, please note that inside [scripts](#) npm will set its own environment variables and Node will prefer those lowercase versions over any uppercase ones that you might set. For details see [this issue](#).

Notice that you need to use underscores instead of dashes, so `--allow-same-version` would become `npm_config_allow_same_version=true`.

npmrc Files

The four relevant files are:

- per-project configuration file (`/path/to/my/project/.npmrc`)
- per-user configuration file (defaults to `$HOME/.npmrc` ; configurable via CLI option `--userconfig` or environment variable `$NPM_CONFIG_USERCONFIG`)
- global configuration file (defaults to `$PREFIX/etc/npmrc` ; configurable via CLI option `--globalconfig` or environment variable `$NPM_CONFIG_GLOBALCONFIG`)
- npm's built-in configuration file (`/path/to/npm/npmrc`)

See [npmrc](#) for more details.

Default Configs

Run `npm config ls -l` to see a set of configuration parameters that are internal to npm, and are defaults if nothing else is specified.

Shorthands and Other CLI Niceties

The following shorthands are parsed on the command-line:

- `-a` : `--all`
- `--enjoy-by` : `--before`
- `-c` : `--call`
- `--desc` : `--description`
- `-f` : `--force`
- `-g` : `--global`
- `-L` : `--location`

- `-d : --loglevel info`
- `-s : --loglevel silent`
- `--silent : --loglevel silent`
- `--ddd : --loglevel silly`
- `--dd : --loglevel verbose`
- `--verbose : --loglevel verbose`
- `-q : --loglevel warn`
- `--quiet : --loglevel warn`
- `-l : --long`
- `-m : --message`
- `--local : --no-global`
- `-n : --no-yes`
- `--no : --no-yes`
- `-p : --parseable`
- `--porcelain : --parseable`
- `-C : --prefix`
- `--readonly : --read-only`
- `--reg : --registry`
- `-S : --save`
- `-B : --save-bundle`
- `-D : --save-dev`
- `-E : --save-exact`
- `-O : --save-optional`
- `-P : --save-prod`
- `-? : --usage`
- `-h : --usage`
- `-H : --usage`
- `--help : --usage`
- `-v : --version`
- `-w : --workspace`
- `--ws : --workspaces`
- `-y : --yes`

If the specified configuration param resolves unambiguously to a known configuration parameter, then it is expanded to that configuration parameter. For example:

```
npm ls --par
# same as:
npm ls --parseable
```

If multiple single-character shorthands are strung together, and the resulting combination is unambiguously not some other configuration param, then it is expanded to its various component pieces. For example:

```
npm ls -gpld
# same as:
npm ls --global --parseable --long --loglevel info
```

Config Settings

`_auth`

- Default: null
- Type: null or String

A basic-auth string to use when authenticating against the npm registry.

Warning: This should generally not be set via a command-line option. It is safer to use a registry-provided authentication bearer token stored in the `~/.npmrc` file by running `npm login`.

`access`

- Default: 'restricted' for scoped packages, 'public' for unscoped packages
- Type: null, "restricted", or "public"

When publishing scoped packages, the access level defaults to `restricted`. If you want your scoped package to be publicly viewable (and installable) set `--access=public`. The only valid values for `access` are `public` and `restricted`. Unscoped packages *always* have an access level of `public`.

Note: Using the `--access` flag on the `npm publish` command will only set the package access level on the initial publish of the package. Any subsequent `npm publish` commands using the `--access` flag will not have an effect to the access level. To make changes to the access level after the initial publish use `npm access`.

`all`

- Default: false
- Type: Boolean

When running `npm outdated` and `npm ls`, setting `--all` will show all outdated or installed packages, rather than only those directly depended upon by the current project.

`allow-same-version`

- Default: false
- Type: Boolean

Prevents throwing an error when `npm version` is used to set the new version to the same value as the current version.

`audit`

- Default: true
- Type: Boolean

When "true" submit audit reports alongside the current npm command to the default registry and all registries configured for scopes. See the documentation for [npm audit](#) for details on what is submitted.

`audit-level`

- Default: null
- Type: null, "info", "low", "moderate", "high", "critical", or "none"

The minimum level of vulnerability for `npm audit` to exit with a non-zero exit code.

`before`

- Default: null

- Type: null or Date

If passed to `npm install`, will rebuild the npm tree such that only versions that were available **on or before** the `--before` time get installed. If there's no versions available for the current set of direct dependencies, the command will error.

If the requested version is a `dist-tag` and the given tag does not pass the `--before` filter, the most recent version less than or equal to that tag will be used. For example, `foo@latest` might install `foo@1.2` even though `latest` is `2.0`.

bin-links

- Default: true
- Type: Boolean

Tells npm to create symlinks (or `.cmd` shims on Windows) for package executables.

Set to false to have it not do this. This can be used to work around the fact that some file systems don't support symlinks, even on ostensibly Unix systems.

browser

- Default: OS X: `"open"`, Windows: `"start"`, Others: `"xdg-open"`
- Type: null, Boolean, or String

The browser that is called by npm commands to open websites.

Set to `false` to suppress browser behavior and instead print urls to terminal.

Set to `true` to use default system URL opener.

ca

- Default: null
- Type: null or String (can be set multiple times)

The Certificate Authority signing certificate that is trusted for SSL connections to the registry. Values should be in PEM format (Windows calls it "Base-64 encoded X.509 (.CER)") with newlines replaced by the string `"\n"`. For example:

```
ca="-----BEGIN CERTIFICATE-----\nXXXX\nXXXX\n-----END CERTIFICATE-----"
```

Set to `null` to only allow "known" registrars, or to a specific CA cert to trust only that specific signing authority.

Multiple CAs can be trusted by specifying an array of certificates:

```
ca["..."]
ca["..."]
```

See also the `strict-ssl` config.

cache

- Default: Windows: `%LocalAppData%\npm-cache`, Posix: `~/.npm`
- Type: Path

The location of npm's cache directory. See [npm cache](#)

cafile

- Default: null
- Type: Path

A path to a file containing one or multiple Certificate Authority signing certificates. Similar to the `ca` setting, but allows for multiple CA's, as well as for the CA information to be stored in a file on disk.

call

- Default: ""
- Type: String

Optional companion option for `npm exec`, `npx` that allows for specifying a custom command to be run along with the installed packages.

```
npm exec --package yo --package generator-node --call "yo node"
```

cert

- Default: null
- Type: null or String

A client certificate to pass when accessing the registry. Values should be in PEM format (Windows calls it "Base-64 encoded X.509 (.CER)") with newlines replaced by the string `"\n"`. For example:

```
cert="-----BEGIN CERTIFICATE-----\nXXXX\nXXXX\n-----END CERTIFICATE-----"
```

It is *not* the path to a certificate file (and there is no "certfile" option).

ci-name

- Default: The name of the current CI system, or `null` when not on a known CI platform.
- Type: null or String

The name of a continuous integration system. If not set explicitly, npm will detect the current CI environment using the [@npmcli/ci-detect](#) module.

cidr

- Default: null
- Type: null or String (can be set multiple times)

This is a list of CIDR address to be used when configuring limited access tokens with the `npm token create` command.

color

- Default: true unless the NO_COLOR environ is set to something other than '0'
- Type: "always" or Boolean

If false, never shows colors. If `"always"` then always shows colors. If true, then only prints color codes for tty file descriptors.

commit-hooks

- Default: true

- Type: Boolean

Run git commit hooks when using the `npm version` command.

depth

- Default: Infinity if `--all` is set, otherwise 1
- Type: null or Number

The depth to go when recursing packages for `npm ls`.

If not set, `npm ls` will show only the immediate dependencies of the root project. If `--all` is set, then npm will show all dependencies by default.

description

- Default: true
- Type: Boolean

Show the description in `npm search`

diff

- Default:
- Type: String (can be set multiple times)

Define arguments to compare in `npm diff`.

diff-dst-prefix

- Default: "b/"
- Type: String

Destination prefix to be used in `npm diff` output.

diff-ignore-all-space

- Default: false
- Type: Boolean

Ignore whitespace when comparing lines in `npm diff`.

diff-name-only

- Default: false
- Type: Boolean

Prints only filenames when using `npm diff`.

diff-no-prefix

- Default: false
- Type: Boolean

Do not show any source or destination prefix in `npm diff` output.

Note: this causes `npm diff` to ignore the `--diff-src-prefix` and `--diff-dst-prefix` configs.

diff-src-prefix

- Default: "a/"

- Type: String

Source prefix to be used in `npm diff` output.

diff-text

- Default: false
- Type: Boolean

Treat all files as text in `npm diff`.

diff-unified

- Default: 3
- Type: Number

The number of lines of context to print in `npm diff`.

dry-run

- Default: false
- Type: Boolean

Indicates that you don't want npm to make any changes and that it should only report what it would have done. This can be passed into any of the commands that modify your local installation, eg, `install`, `update`, `dedupe`, `uninstall`, as well as `pack` and `publish`.

Note: This is NOT honored by other network related commands, eg `dist-tags`, `owner`, etc.

editor

- Default: The EDITOR or VISUAL environment variables, or 'notepad.exe' on Windows, or 'vim' on Unix systems
- Type: String

The command to run for `npm edit` and `npm config edit`.

engine-strict

- Default: false
- Type: Boolean

If set to true, then npm will stubbornly refuse to install (or even consider installing) any package that claims to not be compatible with the current Node.js version.

This can be overridden by setting the `--force` flag.

fetch-retries

- Default: 2
- Type: Number

The "retries" config for the `retry` module to use when fetching packages from the registry.

npm will retry idempotent read requests to the registry in the case of network failures or 5xx HTTP errors.

fetch-retry-factor

- Default: 10
- Type: Number

The "factor" config for the `retry` module to use when fetching packages.

`fetch-retry-maxtimeout`

- Default: 60000 (1 minute)
- Type: Number

The "maxTimeout" config for the `retry` module to use when fetching packages.

`fetch-retry-mintimeout`

- Default: 10000 (10 seconds)
- Type: Number

The "minTimeout" config for the `retry` module to use when fetching packages.

`fetch-timeout`

- Default: 300000 (5 minutes)
- Type: Number

The maximum amount of time to wait for HTTP requests to complete.

`force`

- Default: false
- Type: Boolean

Removes various protections against unfortunate side effects, common mistakes, unnecessary performance degradation, and malicious input.

- Allow clobbering non-npm files in global installs.
- Allow the `npm version` command to work on an unclean git repository.
- Allow deleting the cache folder with `npm cache clean`.
- Allow installing packages that have an `engines` declaration requiring a different version of npm.
- Allow installing packages that have an `engines` declaration requiring a different version of `node`, even if `--engine-strict` is enabled.
- Allow `npm audit fix` to install modules outside your stated dependency range (including SemVer-major changes).
- Allow unpublishing all versions of a published package.
- Allow conflicting peerDependencies to be installed in the root project.
- Implicitly set `--yes` during `npm init`.
- Allow clobbering existing values in `npm pkg`
- Allow unpublishing of entire packages (not just a single version).

If you don't have a clear idea of what you want to do, it is strongly recommended that you do not use this option!

`foreground-scripts`

- Default: false
- Type: Boolean

Run all build scripts (ie, `preinstall`, `install`, and `postinstall`) scripts for installed packages in the foreground process, sharing standard input, output, and error with the main npm process.

Note that this will generally make installs run slower, and be much noisier, but can be useful for debugging.

format-package-lock

- Default: true
- Type: Boolean

Format `package-lock.json` or `npm-shrinkwrap.json` as a human readable file.

fund

- Default: true
- Type: Boolean

When "true" displays the message at the end of each `npm install` acknowledging the number of dependencies looking for funding. See [npm fund](#) for details.

git

- Default: "git"
- Type: String

The command to use for git commands. If git is installed on the computer, but is not in the `PATH`, then set this to the full path to the git binary.

git-tag-version

- Default: true
- Type: Boolean

Tag the commit when using the `npm version` command. Setting this to false results in no commit being made at all.

global

- Default: false
- Type: Boolean

Operates in "global" mode, so that packages are installed into the `prefix` folder instead of the current working directory. See [folders](#) for more on the differences in behavior.

- packages are installed into the `{prefix}/lib/node_modules` folder, instead of the current working directory.
- bin files are linked to `{prefix}/bin`
- man pages are linked to `{prefix}/share/man`

global-style

- Default: false
- Type: Boolean

Causes npm to install the package into your local `node_modules` folder with the same layout it uses with the global `node_modules` folder. Only your direct dependencies will show in `node_modules` and everything they depend on will be flattened in their `node_modules` folders. This obviously will eliminate some deduping. If used with `legacy-bundling`, `legacy-bundling` will be preferred.

globalconfig

- Default: The global `--prefix` setting plus 'etc/npmrc'. For example, '/usr/local/etc/npmrc'
- Type: Path

The config file to read for global config options.

heading

- Default: "npm"
- Type: String

The string that starts all the debugging log output.

https-proxy

- Default: null
- Type: null or URL

A proxy to use for outgoing https requests. If the `HTTPS_PROXY` or `https_proxy` or `HTTP_PROXY` or `http_proxy` environment variables are set, proxy settings will be honored by the underlying `make-fetch-happen` library.

if-present

- Default: false
- Type: Boolean

If true, npm will not exit with an error code when `run-script` is invoked for a script that isn't defined in the `scripts` section of `package.json`. This option can be used when it's desirable to optionally run a script when it's present and fail if the script fails. This is useful, for example, when running scripts that may only apply for some builds in an otherwise generic CI setup.

ignore-scripts

- Default: false
- Type: Boolean

If true, npm does not run scripts specified in package.json files.

Note that commands explicitly intended to run a particular script, such as `npm start`, `npm stop`, `npm restart`, `npm test`, and `npm run-script` will still run their intended script if `ignore-scripts` is set, but they will *not* run any pre- or post-scripts.

include

- Default:
- Type: "prod", "dev", "optional", or "peer" (can be set multiple times)

Option that allows for defining which types of dependencies to install.

This is the inverse of `--omit=<type>`.

Dependency types specified in `--include` will not be omitted, regardless of the order in which omit/include are specified on the command-line.

include-staged

- Default: false
- Type: Boolean

Allow installing "staged" published packages, as defined by [npm RFC PR #92](#).

This is experimental, and not implemented by the npm public registry.

`include-workspace-root`

- Default: false
- Type: Boolean

Include the workspace root when workspaces are enabled for a command.

When false, specifying individual workspaces via the `workspace` config, or all workspaces via the `workspaces` flag, will cause npm to operate only on the specified workspaces, and not on the root project.

`init-author-email`

- Default: ""
- Type: String

The value `npm init` should use by default for the package author's email.

`init-author-name`

- Default: ""
- Type: String

The value `npm init` should use by default for the package author's name.

`init-author-url`

- Default: ""
- Type: "" or URL

The value `npm init` should use by default for the package author's homepage.

`init-license`

- Default: "ISC"
- Type: String

The value `npm init` should use by default for the package license.

`init-module`

- Default: "~/npm-init.js"
- Type: Path

A module that will be loaded by the `npm init` command. See the documentation for the [init-package-json](#) module for more information, or [npm init](#).

`init-version`

- Default: "1.0.0"
- Type: SemVer string

The value that `npm init` should use by default for the package version number, if not already set in package.json.

`json`

- Default: false
- Type: Boolean

Whether or not to output JSON data, rather than the normal output.

- In `npm pkg set` it enables parsing set values with `JSON.parse()` before saving them to your `package.json`.

Not supported by all npm commands.

key

- Default: null
- Type: null or String

A client key to pass when accessing the registry. Values should be in PEM format with newlines replaced by the string `"\n"`. For example:

```
key="-----BEGIN PRIVATE KEY-----\nXXXX\nXXXX\n-----END PRIVATE KEY-----"
```

It is *not* the path to a key file (and there is no "keyfile" option).

legacy-bundling

- Default: false
- Type: Boolean

Causes npm to install the package such that versions of npm prior to 1.4, such as the one included with node 0.8, can install the package. This eliminates all automatic deduping. If used with `global-style` this option will be preferred.

legacy-peer-deps

- Default: false
- Type: Boolean

Causes npm to completely ignore `peerDependencies` when building a package tree, as in npm versions 3 through 6.

If a package cannot be installed because of overly strict `peerDependencies` that collide, it provides a way to move forward resolving the situation.

This differs from `--omit=peer`, in that `--omit=peer` will avoid unpacking `peerDependencies` on disk, but will still design a tree such that `peerDependencies` *could* be unpacked in a correct place.

Use of `legacy-peer-deps` is not recommended, as it will not enforce the `peerDependencies` contract that meta-dependencies may rely on.

link

- Default: false
- Type: Boolean

Used with `npm ls`, limiting output to only those packages that are linked.

local-address

- Default: null
- Type: IP Address

The IP address of the local interface to use when making connections to the npm registry. Must be IPv4 in versions of Node prior to 0.12.

location

- Default: "user" unless `--global` is passed, which will also set this value to "global"
- Type: "global", "user", or "project"

When passed to `npm config` this refers to which config file to use.

lockfile-version

- Default: Version 2 if no lockfile or current lockfile version less than or equal to 2, otherwise maintain current lockfile version
- Type: null, 1, 2, 3, "1", "2", or "3"

Set the lockfile format version to be used in package-lock.json and npm-shrinkwrap.json files. Possible options are:

1: The lockfile version used by npm versions 5 and 6. Lacks some data that is used during the install, resulting in slower and possibly less deterministic installs. Prevents lockfile churn when interoperating with older npm versions.

2: The default lockfile version used by npm version 7. Includes both the version 1 lockfile data and version 3 lockfile data, for maximum determinism and interoperability, at the expense of more bytes on disk.

3: Only the new lockfile information introduced in npm version 7. Smaller on disk than lockfile version 2, but not interoperable with older npm versions. Ideal if all users are on npm version 7 and higher.

loglevel

- Default: "notice"
- Type: "silent", "error", "warn", "notice", "http", "timing", "info", "verbose", or "silly"

What level of logs to report. All logs are written to a debug log, with the path to that file printed if the execution of a command fails.

Any logs of a higher level than the setting are shown. The default is "notice".

See also the `foreground-scripts` config.

logs-max

- Default: 10
- Type: Number

The maximum number of log files to store.

long

- Default: false
- Type: Boolean

Show extended information in `ls`, `search`, and `help-search`.

maxsockets

- Default: 15
- Type: Number

The maximum number of connections to use per origin (protocol/host/port combination).

message

- Default: "%s"

- Type: String

Commit message which is used by `npm version` when creating version commit.

Any "%s" in the message will be replaced with the version number.

node-options

- Default: null
- Type: null or String

Options to pass through to Node.js via the `NODE_OPTIONS` environment variable. This does not impact how npm itself is executed but it does impact how lifecycle scripts are called.

node-version

- Default: Node.js `process.version` value
- Type: SemVer string

The node version to use when checking a package's `engines` setting.

noproxy

- Default: The value of the `NO_PROXY` environment variable
- Type: String (can be set multiple times)

Domain extensions that should bypass any proxies.

Also accepts a comma-delimited string.

npm-version

- Default: Output of `npm --version`
- Type: SemVer string

The npm version to use when checking a package's `engines` setting.

offline

- Default: false
- Type: Boolean

Force offline mode: no network requests will be done during install. To allow the CLI to fill in missing cache data, see

`--prefer-offline` .

omit

- Default: 'dev' if the `NODE_ENV` environment variable is set to 'production', otherwise empty.
- Type: "dev", "optional", or "peer" (can be set multiple times)

Dependency types to omit from the installation tree on disk.

Note that these dependencies *are* still resolved and added to the `package-lock.json` or `npm-shrinkwrap.json` file. They are just not physically installed on disk.

If a package type appears in both the `--include` and `--omit` lists, then it will be included.

If the resulting omit list includes 'dev' , then the `NODE_ENV` environment variable will be set to 'production' for all lifecycle scripts.

`otp`

- Default: null
- Type: null or String

This is a one-time password from a two-factor authenticator. It's needed when publishing or changing package permissions with `npm access`.

If not set, and a registry response fails with a challenge for a one-time password, npm will prompt on the command line for one.

`pack-destination`

- Default: "."
- Type: String

Directory in which `npm pack` will save tarballs.

`package`

- Default:
- Type: String (can be set multiple times)

The package to install for [npm exec](#)

`package-lock`

- Default: true
- Type: Boolean

If set to false, then ignore `package-lock.json` files when installing. This will also prevent *writing* `package-lock.json` if `save` is true.

When package package-locks are disabled, automatic pruning of extraneous modules will also be disabled. To remove extraneous modules with package-locks disabled use `npm prune`.

This configuration does not affect `npm ci`.

`package-lock-only`

- Default: false
- Type: Boolean

If set to true, the current operation will only use the `package-lock.json`, ignoring `node_modules`.

For `update` this means only the `package-lock.json` will be updated, instead of checking `node_modules` and downloading dependencies.

For `list` this means the output will be based on the tree described by the `package-lock.json`, rather than the contents of `node_modules`.

`parseable`

- Default: false
- Type: Boolean

Output parseable results from commands that write to standard output. For `npm search`, this will be tab-separated table format.

prefer-offline

- Default: false
- Type: Boolean

If true, staleness checks for cached data will be bypassed, but missing data will be requested from the server. To force full offline mode, use `--offline`.

prefer-online

- Default: false
- Type: Boolean

If true, staleness checks for cached data will be forced, making the CLI look for updates immediately even for fresh package data.

prefix

- Default: In global mode, the folder where the node executable is installed. In local mode, the nearest parent folder containing either a package.json file or a node_modules folder.
- Type: Path

The location to install global items. If set on the command line, then it forces non-global commands to run in the specified folder.

preid

- Default: ""
- Type: String

The "prerelease identifier" to use as a prefix for the "prerelease" part of a semver. Like the `rc` in `1.2.0-rc.8`.

progress

- Default: `true` unless running in a known CI system
- Type: Boolean

When set to `true`, npm will display a progress bar during time intensive operations, if `process.stderr` is a TTY.

Set to `false` to suppress the progress bar.

proxy

- Default: null
- Type: null, false, or URL

A proxy to use for outgoing http requests. If the `HTTP_PROXY` or `http_proxy` environment variables are set, proxy settings will be honored by the underlying `request` library.

read-only

- Default: false
- Type: Boolean

This is used to mark a token as unable to publish when configuring limited access tokens with the `npm token create` command.

rebuild-bundle

- Default: true
- Type: Boolean

Rebuild bundled dependencies after installation.

registry

- Default: "<https://registry.npmjs.org/>"
- Type: URL

The base URL of the npm registry.

save

- Default: `true` unless when using `npm update` or `npm dedupe` where it defaults to `false`
- Type: Boolean

Save installed packages to a `package.json` file as dependencies.

When used with the `npm rm` command, removes the dependency from `package.json`.

Will also prevent writing to `package-lock.json` if set to `false`.

save-bundle

- Default: false
- Type: Boolean

If a package would be saved at install time by the use of `--save`, `--save-dev`, or `--save-optional`, then also put it in the `bundleDependencies` list.

Ignored if `--save-peer` is set, since `peerDependencies` cannot be bundled.

save-dev

- Default: false
- Type: Boolean

Save installed packages to a `package.json` file as `devDependencies`.

save-exact

- Default: false
- Type: Boolean

Dependencies saved to `package.json` will be configured with an exact version rather than using npm's default semver range operator.

save-optional

- Default: false
- Type: Boolean

Save installed packages to a `package.json` file as `optionalDependencies`.

save-peer

- Default: false
- Type: Boolean

Save installed packages to a `package.json` file as `peerDependencies`.

save-prefix

- Default: "^"
- Type: String

Configure how versions of packages installed to a package.json file via `--save` or `--save-dev` get prefixed.

For example if a package has version `1.2.3`, by default its version is set to `^1.2.3` which allows minor upgrades for that package, but after `npm config set save-prefix='~'` it would be set to `~1.2.3` which only allows patch upgrades.

save-prod

- Default: false
- Type: Boolean

Save installed packages into `dependencies` specifically. This is useful if a package already exists in `devDependencies` or `optionalDependencies`, but you want to move it to be a non-optional production dependency.

This is the default behavior if `--save` is true, and neither `--save-dev` or `--save-optional` are true.

scope

- Default: the scope of the current project, if any, or ""
- Type: String

Associate an operation with a scope for a scoped registry.

Useful when logging in to or out of a private registry:

```
# log in, linking the scope to the custom registry
npm login --scope=@mycorp --registry=https://registry.mycorp.com

# log out, removing the link and the auth token
npm logout --scope=@mycorp
```

This will cause `@mycorp` to be mapped to the registry for future installation of packages specified according to the pattern `@mycorp/package`.

This will also cause `npm init` to create a scoped package.

```
# accept all defaults, and create a package named "@foo/whatever",
# instead of just named "whatever"
npm init --scope=@foo --yes
```

script-shell

- Default: `'/bin/sh'` on POSIX systems, `'cmd.exe'` on Windows
- Type: null or String

The shell to use for scripts run with the `npm exec`, `npm run` and `npm init <pkg>` commands.

searchexclude

- Default: ""
- Type: String

Space-separated options that limit the results from search.

searchlimit

- Default: 20
- Type: Number

Number of items to limit search results to. Will not apply at all to legacy searches.

searchopts

- Default: ""
- Type: String

Space-separated options that are always passed to search.

searchstaleness

- Default: 900
- Type: Number

The age of the cache, in seconds, before another registry request is made if using legacy search endpoint.

shell

- Default: SHELL environment variable, or "bash" on Posix, or "cmd.exe" on Windows
- Type: String

The shell to run for the `npm explore` command.

sign-git-commit

- Default: false
- Type: Boolean

If set to true, then the `npm version` command will commit the new package version using `-s` to add a signature.

Note that git requires you to have set up GPG keys in your git configs for this to work properly.

sign-git-tag

- Default: false
- Type: Boolean

If set to true, then the `npm version` command will tag the version using `-s` to add a signature.

Note that git requires you to have set up GPG keys in your git configs for this to work properly.

strict-peer-deps

- Default: false
- Type: Boolean

If set to `true`, and `--legacy-peer-deps` is not set, then *any* conflicting `peerDependencies` will be treated as an install failure, even if npm could reasonably guess the appropriate resolution based on non-peer dependency relationships.

By default, conflicting `peerDependencies` deep in the dependency graph will be resolved using the nearest non-peer dependency specification, even if doing so will result in some packages receiving a peer dependency outside

the range set in their package's `peerDependencies` object.

When such an override is performed, a warning is printed, explaining the conflict and the packages involved. If `--strict-peer-deps` is set, then this warning is treated as a failure.

`strict-ssl`

- Default: true
- Type: Boolean

Whether or not to do SSL key validation when making requests to the registry via https.

See also the `ca` config.

`tag`

- Default: "latest"
- Type: String

If you ask npm to install a package and don't tell it a specific version, then it will install the specified tag.

Also the tag that is added to the package@version specified by the `npm tag` command, if no explicit tag is given.

When used by the `npm diff` command, this is the tag used to fetch the tarball that will be compared with the local files by default.

`tag-version-prefix`

- Default: "v"
- Type: String

If set, alters the prefix used when tagging a new version when performing a version increment using `npm-version`. To remove the prefix altogether, set it to the empty string: `""`.

Because other tools may rely on the convention that npm version tags look like `v1.0.0`, *only use this property if it is absolutely necessary*. In particular, use care when overriding this setting for public packages.

`timing`

- Default: false
- Type: Boolean

If true, writes an `npm-debug` log to `_logs` and timing information to `_timing.json`, both in your cache, even if the command completes successfully. `_timing.json` is a newline delimited list of JSON objects.

You can quickly view it with this [json](#) command line: `npm exec -- json -g < ~/.npm/_timing.json`.

`umask`

- Default: 0
- Type: Octal numeric string in range 0000..0777 (0..511)

The "umask" value to use when setting the file creation mode on files and folders.

Folders and executables are given a mode which is `0o777` masked against this value. Other files are given a mode which is `0o666` masked against this value.

Note that the underlying system will *also* apply its own umask value to files and folders that are created, and npm does not circumvent this, but rather adds the `--umask` config to it.

Thus, the effective default umask value on most POSIX systems is 0o22, meaning that folders and executables are created with a mode of 0o755 and other files are created with a mode of 0o644.

unicode

- Default: false on windows, true on mac/unix systems with a unicode locale, as defined by the `LC_ALL`, `LC_CTYPE`, or `LANG` environment variables.
- Type: Boolean

When set to true, npm uses unicode characters in the tree output. When false, it uses ascii characters instead of unicode glyphs.

update-notifier

- Default: true
- Type: Boolean

Set to false to suppress the update notification when using an older version of npm than the latest.

usage

- Default: false
- Type: Boolean

Show short usage output about the command specified.

user-agent

- Default: "npm/{npm-version} node/{node-version} {platform} {arch} workspaces/{workspaces} {ci}"
- Type: String

Sets the User-Agent request header. The following fields are replaced with their actual counterparts:

- `{npm-version}` - The npm version in use
- `{node-version}` - The Node.js version in use
- `{platform}` - The value of `process.platform`
- `{arch}` - The value of `process.arch`
- `{workspaces}` - Set to `true` if the `workspaces` or `workspace` options are set.
- `{ci}` - The value of the `ci-name` config, if set, prefixed with `ci/`, or an empty string if `ci-name` is empty.

userconfig

- Default: "~/.npmrc"
- Type: Path

The location of user-level configuration settings.

This may be overridden by the `npm_config_userconfig` environment variable or the `--userconfig` command line option, but may *not* be overridden by settings in the `globalconfig` file.

version

- Default: false
- Type: Boolean

If true, output the npm version and exit successfully.

Only relevant when specified explicitly on the command line.

versions

- Default: false
- Type: Boolean

If true, output the npm version as well as node's `process.versions` map and the version in the current working directory's `package.json` file if one exists, and exit successfully.

Only relevant when specified explicitly on the command line.

viewer

- Default: "man" on Posix, "browser" on Windows
- Type: String

The program to use to view help content.

Set to `"browser"` to view html help content in the default web browser.

which

- Default: null
- Type: null or Number

If there are multiple funding sources, which 1-indexed source URL to open.

workspace

- Default:
- Type: String (can be set multiple times)

Enable running a command in the context of the configured workspaces of the current project while filtering by running only the workspaces defined by this configuration option.

Valid values for the `workspace` config are either:

- Workspace names
- Path to a workspace directory
- Path to a parent workspace directory (will result in selecting all workspaces within that folder)

When set for the `npm init` command, this may be set to the folder of a workspace which does not yet exist, to create the folder and set it up as a brand new workspace within the project.

This value is not exported to the environment for child processes.

workspaces

- Default: null
- Type: null or Boolean

Set to true to run the command in the context of **all** configured workspaces.

Explicitly setting this to false will cause commands like `install` to ignore workspaces altogether. When not set explicitly:

- Commands that operate on the `node_modules` tree (install, update, etc.) will link workspaces into the `node_modules` folder. - Commands that do other things (test, exec, publish, etc.) will operate on the root project, *unless* one or more workspaces are specified in the `workspace` config.

This value is not exported to the environment for child processes.

yes

- Default: null
- Type: null or Boolean

Automatically answer "yes" to any prompts that npm might print on the command line.

also

- Default: null
- Type: null, "dev", or "development"
- DEPRECATED: Please use `--include=dev` instead.

When set to `dev` or `development`, this is an alias for `--include=dev`.

auth-type

- Default: "legacy"
- Type: "legacy", "sso", "saml", or "oauth"
- DEPRECATED: This method of SSO/SAML/OAuth is deprecated and will be removed in a future version of npm in favor of web-based login.

What authentication strategy to use with `adduser` / `login`.

cache-max

- Default: Infinity
- Type: Number
- DEPRECATED: This option has been deprecated in favor of `--prefer-online`

`--cache-max=0` is an alias for `--prefer-online`

cache-min

- Default: 0
- Type: Number
- DEPRECATED: This option has been deprecated in favor of `--prefer-offline`.

`--cache-min=9999` (or bigger) is an alias for `--prefer-offline`.

dev

- Default: false
- Type: Boolean
- DEPRECATED: Please use `--include=dev` instead.

Alias for `--include=dev`.

init.author.email

- Default: ""
- Type: String
- DEPRECATED: Use `--init-author-email` instead.

Alias for `--init-author-email`

init.author.name

- Default: ""
- Type: String

- DEPRECATED: Use `--init-author-name` instead.

Alias for `--init-author-name`

`init.author.url`

- Default: ""
- Type: "" or URL
- DEPRECATED: Use `--init-author-url` instead.

Alias for `--init-author-url`

`init.license`

- Default: "ISC"
- Type: String
- DEPRECATED: Use `--init-license` instead.

Alias for `--init-license`

`init.module`

- Default: "~/npm-init.js"
- Type: Path
- DEPRECATED: Use `--init-module` instead.

Alias for `--init-module`

`init.version`

- Default: "1.0.0"
- Type: SemVer string
- DEPRECATED: Use `--init-version` instead.

Alias for `--init-version`

`only`

- Default: null
- Type: null, "prod", or "production"
- DEPRECATED: Use `--omit=dev` to omit dev dependencies from the install.

When set to `prod` or `production`, this is an alias for `--omit=dev`.

`optional`

- Default: null
- Type: null or Boolean
- DEPRECATED: Use `--omit=optional` to exclude optional dependencies, or `--include=optional` to include them.

Default value does install optional deps unless otherwise omitted.

Alias for `--include=optional` or `--omit=optional`

`production`

- Default: null
- Type: null or Boolean

- DEPRECATED: Use `--omit=dev` instead.

Alias for `--omit=dev`

shrinkwrap

- Default: true
- Type: Boolean
- DEPRECATED: Use the `--package-lock` setting instead.

Alias for `--package-lock`

sso-poll-frequency

- Default: 500
- Type: Number
- DEPRECATED: The `--auth-type` method of SSO/SAML/OAuth will be removed in a future version of npm in favor of web-based login.

When used with SSO-enabled `auth-type` s, configures how regularly the registry should be polled while the user is completing authentication.

sso-type

- Default: "oauth"
- Type: null, "oauth", or "saml"
- DEPRECATED: The `--auth-type` method of SSO/SAML/OAuth will be removed in a future version of npm in favor of web-based login.

If `--auth-type=sso` , the type of SSO type to use.

tmp

- Default: The value returned by the Node.js `os.tmpdir()` method https://nodejs.org/api/os.html#os_os_tmpdir
- Type: Path
- DEPRECATED: This setting is no longer used. npm stores temporary files in a special location in the cache, and they are managed by [cacache](#) .

Historically, the location where temporary files were stored. No longer relevant.

See also

- [npm config](#)
- [npmrc](#)
- [npm scripts](#)
- [npm folders](#)
- [npm](#)