

npm-config

More than you probably want to know about npm configuration

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 `npm-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:

- `-v` : `--version`
- `-h` , `-?` , `--help` , `-H` : `--usage`
- `-s` , `--silent` : `--loglevel silent`
- `-q` , `--quiet` : `--loglevel warn`
- `-d` : `--loglevel info`
- `-dd` , `--verbose` : `--loglevel verbose`
- `-ddd` : `--loglevel silly`
- `-g` : `--global`
- `-C` : `--prefix`
- `-l` : `--long`
- `-m` : `--message`
- `-p` , `--porcelain` : `--parseable`
- `-reg` : `--registry`
- `-f` : `--force`
- `-desc` : `--description`
- `-S` : `--save`
- `-P` : `--save-prod`
- `-D` : `--save-dev`
- `-O` : `--save-optional`
- `-B` : `--save-bundle`
- `-E` : `--save-exact`
- `-y` : `--yes`
- `-n` : `--yes false`
- `ll` and `la` commands: `ls --long`

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
```

Per-Package Config Settings

When running scripts (see [npm-scripts](#)) the package.json “config” keys are overwritten in the environment if there is a config param of `<name>[@<version>]:<key>`. For example, if the package.json has this:

```
{ "name" : "foo"
, "config" : { "port" : "8080" }
, "scripts" : { "start" : "node server.js" } }
```

and the server.js is this:

```
http.createServer(...).listen(process.env.npm_package_config_port)
```

then the user could change the behavior by doing:

```
npm config set foo:port 80
```

See [package.json](#) for more information.

Config Settings

access

- Default: **restricted**
- Type: Access

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` .

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.

always-auth

- Default: false
- Type: Boolean

Force npm to always require authentication when accessing the registry, even for `GET` requests.

also

- Default: null
- Type: String

When “dev” or “development” and running local `npm shrinkwrap` , `npm outdated` , or `npm update` , is an alias for `--dev` .

audit

- Default: true
- Type: Boolean

When “true” submit audit reports alongside `npm install` runs 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: "low"
- Type: 'low' , 'moderate' , 'high' , 'critical'

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

auth-type

- Default: `'legacy'`
- Type: `'legacy'` , `'sso'` , `'saml'` , `'oauth'`

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

before

- Alias: `enjoy-by`
- Default: `null`
- Type: `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: `String`

The browser that is called by the `npm docs` command to open websites.

ca

- Default: The npm CA certificate
- Type: `String`, `Array` or `null`

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.

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.

cache

- Default: Windows: `%AppData%\npm-cache` , Posix: `~/.npm`
- Type: path

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

cache-lock-stale

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

The number of ms before cache folder lockfiles are considered stale.

cache-lock-retries

- Default: 10
- Type: Number

Number of times to retry to acquire a lock on cache folder lockfiles.

cache-lock-wait

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

Number of ms to wait for cache lock files to expire.

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: 10
- Type: Number

DEPRECATED: This option has been deprecated in favor of `--prefer-offline` .

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

cert

- Default: `null`
- Type: 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).

cidr

- Default: `null`
- Type: String, Array, null

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

color

- Default: true
- Type: Boolean or "always"

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

This option can also be changed using the environment: colors are disabled when the environment variable `NO_COLOR` is set to any value.

depth

- Default: Infinity
- Type: Number

The depth to go when recursing directories for `npm ls`, `npm cache ls`, and `npm outdated`.

For `npm outdated`, a setting of `Infinity` will be treated as `0` since that gives more useful information. To show the outdated status of all packages and dependents, use a large integer value, e.g., `npm outdated --depth 9999`

description

- Default: true
- Type: Boolean

Show the description in `npm search`

dev

- Default: false
- Type: Boolean

Install `dev-dependencies` along with packages.

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` . This is NOT currently honored by some network related commands, eg `dist-tags` , `owner` , etc.

editor

- Default: `EDITOR` environment variable if set, or `"vi"` on Posix, or `"notepad"` on Windows.
- Type: path

The command to run for `npm edit` or `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.

force

- Default: false
- Type: Boolean

Makes various commands more forceful.

- lifecycle script failure does not block progress.
- publishing clobbers previously published versions.
- skips cache when requesting from the registry.
- prevents checks against clobbering non-npm files.

format-package-lock

- Default: true
- Type: Boolean

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

fetch-retries

- Default: 2

- Type: Number

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

fetch-retry-factor

- Default: 10
- Type: Number

The “factor” 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-retry-maxtimeout

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

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

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.

commit-hooks

- Default: `true`
- Type: Boolean

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

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 [npm-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`

globalconfig

- Default: `{prefix}/etc/npmrc`
- Type: path

The config file to read for global config options.

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.

group

- Default: GID of the current process
- Type: String or Number

The group to use when running package scripts in global mode as the root user.

heading

- Default: "npm"

- Type: String

The string that starts all the debugging log output.

https-proxy

- Default: null
- Type: 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 `request` 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-prepublish

- Default: false
- Type: Boolean

If true, npm will not run `prepublish` scripts.

ignore-scripts

- Default: false
- Type: Boolean

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

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-author-name

- Default: ""
- Type: String

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

init-author-email

- Default: ""
- Type: String

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

init-author-url

- Default: ""
- Type: String

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-version

- Default: "1.0.0"
- Type: semver

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.

This feature is currently experimental, and the output data structures for many commands is either not implemented in JSON yet, or subject to change. Only the output from `npm ls --json` and `npm search --json` are currently valid.

key

- Default: `null`
- Type: 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.

link

- Default: `false`
- Type: Boolean

If true, then local installs will link if there is a suitable globally installed package.

Note that this means that local installs can cause things to be installed into the global space at the same time. The link is only done if one of the two conditions are met:

- The package is not already installed globally, or
- the globally installed version is identical to the version that is being installed locally.

local-address

- Default: `undefined`
- 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.

loglevel

- Default: “notice”
- Type: String
- Values: “silent”, “error”, “warn”, “notice”, “http”, “timing”, “info”, “verbose”, “silly”

What level of logs to report. On failure, *all* logs are written to **npm-debug.log** in the current working directory.

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

logstream

- Default: process.stderr
- Type: Stream

This is the stream that is passed to the **npmlog** module at run time.

It cannot be set from the command line, but if you are using npm programmatically, you may wish to send logs to somewhere other than stderr.

If the **color** config is set to true, then this stream will receive colored output if it is a TTY.

logs-max

- Default: 10
- Type: Number

The maximum number of log files to store.

long

- Default: false
- Type: Boolean

Show extended information in **npm ls** and **npm search**.

maxsockets

- Default: 50
- Type: Number

The maximum number of connections to use per origin (protocol/host/port combination). Passed to the `http Agent` used to make the request.

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.

metrics-registry

- Default: The value of `registry` (which defaults to “https://registry.npmjs.org/”)
- Type: String

The registry you want to send cli metrics to if `send-metrics` is true.

node-options

- Default: null
- Type: 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: process.version
- Type: semver or false

The node version to use when checking a package’s `engines` map.

noproxy

- Default: null
- Type: String or Array

A comma-separated string or an array of domain extensions that a proxy should not be used for.

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` .

onload-script

- Default: false
- Type: path

A node module to `require()` when npm loads. Useful for programmatic usage.

only

- Default: null
- Type: String

When “dev” or “development” and running local `npm install` without any arguments, only `devDependencies` (and their dependencies) are installed.

When “dev” or “development” and running local `npm ls` , `npm outdated` , or `npm update` , is an alias for `--dev` .

When “prod” or “production” and running local `npm install` without any arguments, only non-`devDependencies` (and their dependencies) are installed.

When “prod” or “production” and running local `npm ls` , `npm outdated` , or `npm update` , is an alias for `--production` .

optional

- Default: true
- Type: Boolean

Attempt to install packages in the `optionalDependencies` object. Note that if these packages fail to install, the overall installation process is not aborted.

otp

- Default: null
- Type: Number

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

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 option is an alias for `--shrinkwrap` .

package-lock-only

- Default: false
- Type: Boolean

If set to true, it will update only the `package-lock.json` , instead of checking `node_modules` and downloading dependencies.

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` .

This option is effectively equivalent to `--cache-min=9999999` .

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: see `npm-folders`
- 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`.

production

- Default: false
- Type: Boolean

Set to true to run in “production” mode.

1. devDependencies are not installed at the topmost level when running local `npm install` without any arguments.
2. Set the `NODE_ENV="production"` for lifecycle scripts.

progress

- Default: true, unless TRAVIS or CI env vars set.
- 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: 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 package registry.

rollback

- Default: true
- Type: Boolean

Remove failed installs.

save

- Default: true
- Type: Boolean

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

When used with the `npm rm` command, it removes it from the `dependencies` object.

Only works if there is already a package.json file present.

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.

When used with the `npm rm` command, it removes it from the `bundleDependencies` list.

save-prod

- Default: false
- Type: Boolean

Makes sure that a package will be saved into `dependencies` specifically. This is useful if a package already exists in `devDependencies` or `optionalDependencies` , but you want to move it to be a production dep. This is also the default behavior if `--save` is true, and neither `--save-dev` or `--save-optional` are true.

save-dev

- Default: false
- Type: Boolean

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

When used with the `npm rm` command, it removes it from the `devDependencies` object.

Only works if there is already a package.json file present.

save-exact

- Default: false
- Type: Boolean

Dependencies saved to package.json using `--save` , `--save-dev` or `--save-optional` 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`.

When used with the `npm rm` command, it removes it from the `devDependencies` object.

Only works if there is already a `package.json` file present.

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.

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 a private registry for the first time: `npm login --scope=@organization --registry=registry.organization.com`, which will cause `@organization` to be mapped to the registry for future installation of packages specified according to the pattern `@organization/package`.

script-shell

- Default: `null`
- Type: path

The shell to use for scripts run with the `npm run` command.

scripts-prepend-node-path

- Default: `"warn-only"`
- Type: Boolean, `"auto"` or `"warn-only"`

If set to `true`, add the directory in which the current `node` executable resides to the `PATH` environment variable when running scripts, even if that means that `npm` will invoke a different `node` executable than the one which it is running.

If set to `false`, never modify `PATH` with that.

If set to **"warn-only"** , never modify **PATH** but print a warning if **npm** thinks that you may want to run it with **true** , e.g. because the **node** executable in the **PATH** is not the one **npm** was invoked with.

If set to **auto** , only add that directory to the **PATH** environment variable if the **node** executable with which **npm** was invoked and the one that is found first on the **PATH** are different.

searchexclude

- Default: ""
- Type: String

Space-separated options that limit the results from search.

searchopts

- Default: ""
- Type: String

Space-separated options that are always passed to search.

searchlimit

- Default: 20
- Type: Number

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

searchstaleness

- Default: 900 (15 minutes)
- Type: Number

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

send-metrics

- Default: false
- Type: Boolean

If true, success/failure metrics will be reported to the registry stored in **metrics-registry** . These requests contain the number of successful and failing runs of the npm CLI and the time

period over which those counts were gathered. No identifying information is included in these requests.

shell

- Default: SHELL environment variable, or “bash” on Posix, or “cmd” on Windows
- Type: path

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

shrinkwrap

- Default: true
- Type: Boolean

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

This option is an alias for `--package-lock`.

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.

sso-poll-frequency

- Default: 500
- Type: Number

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: 'oauth', 'saml', or null

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

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.

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. `_timing.json` is a newline delimited list of JSON objects. You can quickly view it with this `json` command line: `json -g < ~/.npm/_timing.json`.

tmp

- Default: `TMPDIR` environment variable, or `"/tmp"`
- Type: path

Where to store temporary files and folders. All temp files are deleted on success, but left behind on failure for forensic purposes.

unicode

- Default: false on windows, true on mac/unix systems with a unicode locale
- Type: Boolean

When set to true, npm uses unicode characters in the tree output. When false, it uses ascii characters to draw trees.

unsafe-perm

- Default: false if running as root, true otherwise
- Type: Boolean

Set to true to suppress the UID/GID switching when running package scripts. If set explicitly to false, then installing as a non-root user will fail.

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

Set to show short usage output (like the -H output) instead of complete help when doing **npm-help** .

user

- Default: “nobody”
- Type: String or Number

The UID to set to when running package scripts as root.

userconfig

- Default: ~/.npmrc
- Type: path

The location of user-level configuration settings.

umask

- Default: 022
- 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 **0777** masked against this value. Other files are given a mode which is **0666** masked against this value. Thus, the defaults are **0755** and **0644** respectively.

user-agent

- Default: node/{process.version} {process.platform} {process.arch}
- Type: String

Sets a User-Agent to the request header

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 exit successfully.

Only relevant when specified explicitly on the command line.

viewer

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

The program to use to view help content.

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