



Linux Ubuntu 26.04 Manual Pages on command 'nyc.1'

\$ man nyc.1

NYC.JS(1) User Commands NYC.JS(1)

NAME

nyc.js - istanbul command line interface

DESCRIPTION

nyc.js [command] [options] nyc.js [options] [bin-to-instrument]

Commands:

nyc.js check-coverage

check whether coverage is within thresholds provided

nyc.js instrument <input> [output]

instruments a file or a directory tree and writes the instrumented code to the de?

sired output location

nyc.js report

run coverage report for .nyc_output

nyc.js merge <input-directory>

merge istanbul format coverage

[output-file]

output in a given folder

OPTIONS

--cwd working directory used when resolving paths [string] [default: "."]

--nycrc-path

specify an explicit path to find nyc configuration

-t, --temp-dir

directory to output raw coverage information to [string] [default:

"/.nyc_output"]

-x, --exclude

a list of specific files and directories that should be excluded from coverage, glob

patterns are supported [string] [default:

```
["coverage/**","packages/*/test{,s}/**","**/*.d.ts","test{,s}/**","test{,-*}.{  
js,cjs,mjs,ts,tsx,jsx}","**/*{.,-}test.{js,cjs,mjs,ts,tsx,jsx}","**/__tests__/  
**","**/{ava,babel,nyc}.config.{js,cjs,mjs}","**/jest.config.{js,cjs,mjs,ts}",  
"**/{karma,rollup,webpack}.config.js","**/.{eslint,mocha}rc.{js,cjs}"]]
```

--exclude-node-modules

whether or not to exclude all node_module folders (i.e. **/node_modules/**) by de?

fault

[boolean] [default: true]

-n, --include

a list of specific files that should be covered, glob patterns are supported

[string] [default: []]

-e, --extension

a list of extensions that nyc should handle in addition to .js

[string] [default: [".js",".cjs",".mjs",".ts",".tsx",".jsx"]]

--ignore-class-methods

class method names to ignore for coverage [string] [default: []]

--auto-wrap

allow `return` statements outside of functions [boolean] [default: true]

--es-modules

should files be treated as ES Modules [boolean] [default: true]

--parser-plugins

babel parser plugins to use when parsing the source

[string] [default: ["asyncGenerators","bigInt","classProperties","classPrivate Prop?
erties","classPrivateMethods","dynamicImport","importMeta","numericSeparat
or","objectRestSpread","optionalCatchBinding","topLevelAwait"]]

--compact

should the output be compacted? [boolean] [default: true]

--preserve-comments

should comments be preserved in the output? [boolean] [default: true]

--produce-source-map

should source maps be produced? [boolean] [default: true]

--source-map

should nyc detect and handle source maps? [boolean] [default: true]

-i, --require

a list of additional modules that nyc should attempt to require in its subprocess,

e.g., @babel/register, @babel/polyfill

[string] [default: []]

--instrument

should nyc handle instrumentation? [boolean] [default: true]

--exclude-after-remap

should exclude logic be performed after the source-map remaps filenames?

[boolean] [default: true]

--branches

what % of branches must be covered? [number] [default: 0]

--functions

what % of functions must be covered? [number] [default: 0]

--lines

what % of lines must be covered? [number] [default: 90]

--statements

what % of statements must be covered? [number] [default: 0]

--per-file

check thresholds per file [boolean] [default: false]

--check-coverage

check whether coverage is within thresholds provided [boolean] [default:

false]

-r, --reporter

coverage reporter(s) to use [string] [default: ["text"]]

--report-dir

directory to output coverage reports in [string] [default: "coverage"]

--show-process-tree

display the tree of spawned processes [boolean] [default: false]

--skip-empty

don't show empty files (no lines of code) in report [boolean] [default: false]

--skip-full

don't show files with 100% statement, branch, and function coverage [boolean] [default: false]

-s, --silent

don't output a report after tests finish running [boolean] [default: false]

-a, --all

whether or not to instrument all files of the project (not just the ones touched by your test suite) [boolean] [default: false]

--eager

instantiate the instrumenter at startup (see <https://git.io/vMKZ9>) [boolean] [default: false]

-c, --cache

cache instrumentation results for improved performance [boolean] [default: true]

--cache-dir

explicitly set location for instrumentation cache [string]

--babel-cache

cache babel transpilation results for improved performance [boolean] [default: false]

--use-spawn-wrap

use spawn-wrap instead of setting process.env.NODE_OPTIONS [boolean] [default: false]

--hook-require

should nyc wrap require? [boolean] [default: true]

--hook-run-in-context

should nyc wrap vm.runInContext? [boolean] [default: false]

--hook-run-in-this-context

should nyc wrap vm.runInThisContext? [boolean] [default: false]

--clean

should the .nyc_output folder be cleaned before executing tests

[boolean] [default: true]

-h, --help

Show help [boolean]

--version

Show version number [boolean]

EXAMPLES

nyc.js npm test

instrument your tests with coverage

nyc.js --require @babel/register npm

instrument your tests with coverage

test and transpile with Babel

nyc.js report --reporter=text-lcov

output lcov report after running your tests

visit <https://git.io/vHysA> for list of available reporters

nyc.js 15.1.0

January 2026

NYC.JS(1)