



## ***Rocky Enterprise Linux 9.2 Manual Pages on command 'libnpx.1'***

**C:\>man libnpx.1**

NPX(1) User Commands NPX(1)

### NAME

npx - execute npm package binaries

### SYNOPSIS

npx [options] <command>[@version] [command-arg]...

npx [options] [-p|--package <pkg>]... <command> [command-arg]...

npx [options] -c '<command-string>'

npx --shell-auto-fallback [shell]

### INSTALL

npm install -g npx

### DESCRIPTION

Executes <command> either from a local node\_modules/.bin, or from a central cache, installing any packages needed in order for <command> to run.

By default, npx will check whether <command> exists in \$PATH, or in the local project binaries, and execute that. If <command> is not found, it will be installed prior to execution.

Unless a --package option is specified, npx will try to guess the name of the binary to invoke depending on the specifier provided. All package specifiers understood by npm may be used with npx, including git specifiers, remote tarballs, local directories, or scoped packages.

If a full specifier is included, or if --package is used, npx will always use a freshly-installed, temporary version of the package. This can also be forced with

the `--ignore-existing` flag.

? `-p, --package <package>` - define the package to be installed. This defaults to the value of `<command>`. This is only needed for packages with multiple binaries if you want to call one of the other executables, or where the binary name does not match the package name. If this option is provided `<command>` will be executed as-is, without interpreting `@version` if it's there. Multiple `--package` options may be provided, and all the packages specified will be installed.

? `--no-install` - If passed to `npx`, it will only try to run `<command>` if it already exists in the current path or in `$prefix/node_modules/.bin`. It won't try to install missing commands.

? `--cache <path>` - set the location of the `npm` cache. Defaults to `npm`'s own cache settings.

? `--userconfig <path>` - path to the user configuration file to pass to `npm`. Defaults to whatever `npm`'s current default is.

? `-c <string>` - Execute `<string>` inside an `npm` run-script-like shell environment, with all the usual environment variables available. Only the first item in `<string>` will be automatically used as `<command>`. Any others must use `-p`.

? `--shell <string>` - The shell to invoke the command with, if any.

? `--shell-auto-fallback [<shell>]` - Generates shell code to override your shell's "command not found" handler with one that calls `npx`. Tries to figure out your shell, or you can pass its name (either `bash`, `fish`, or `zsh`) as an option. See below for how to install.

? `--ignore-existing` - If this flag is set, `npx` will not look in `$PATH`, or in the current package's `node_modules/.bin` for an existing version before deciding whether to install. Binaries in those paths will still be available for execution, but will be shadowed by any packages requested by this install.

? `-q, --quiet` - Suppress any output from `npx` itself (progress bars, error messages, install reports). Subcommand output itself will not be silenced.

? `-n, --node-arg` - Extra node argument to supply to `node` when binary is a node script. You can supply this option multiple times to add more arguments.

? `-v, --version` - Show the current `npx` version.

## EXAMPLES

Running a project-local bin

```
$ npm i -D webpack
```

```
$ npx webpack ...
```

One-off invocation without local installation

```
$ npm rm webpack
```

```
$ npx webpack -- ...
```

```
$ cat package.json
```

```
...webpack not in "devDependencies"...
```

Invoking a command from a github repository

```
$ npx github:piuccio/cowsay
```

```
...or...
```

```
$ npx git+ssh://my.hosted.git:cowsay.git#semver:^1
```

```
...etc...
```

Execute a full shell command using one npx call w/ multiple packages

```
$ npx -p lolcatjs -p cowsay -c \
```

```
'echo "$npm_package_name@$npm_package_version" | cowsay | lolcatjs'
```

```
...
```

```
_____
```

```
< your-cool-package@1.2.3 >
```

```
-----
```

```
\ ^__^
```

```
\ (oo)\_____
```

```
(__)\       )\/
```

```
||----w |
```

```
||  ||
```

Run node binary with --inspect

```
$ npx --node-arg=--inspect cowsay
```

```
Debugger listening on ws://127.0.0.1:9229/...
```

Specify a node version to run npm scripts (or anything else!)

```
npx -p node@8 npm run build
```

## SHELL AUTO FALLBACK

You can configure npx to run as your default fallback command when you type some?

thing in the command line with an @ but the command is not found. This includes in?

stalling packages that were not found in the local prefix either.

For example:

```
$ npm@4 --version
```

```
(stderr) npm@4 not found. Trying with npx...
```

```
4.6.1
```

```
$ asdfasdf
```

```
zsh: command not found: asdfasdf
```

Currently, zsh, bash ( $\geq 4$ ), and fish are supported. You can access these completion scripts using `npx --shell-auto-fallback <shell>`.

To install permanently, add the relevant line below to your `~/.bashrc`, `~/.zshrc`, `~/.config/fish/config.fish`, or as needed. To install just for the shell session, simply run the line.

You can optionally pass through `--no-install` when generating the fallback to prevent it from installing packages if the command is missing.

For bash  $\geq 4$ :

```
$ source <(npx --shell-auto-fallback bash)
```

For zsh:

```
$ source <(npx --shell-auto-fallback zsh)
```

For fish:

```
$ source (npx --shell-auto-fallback fish | psub)
```

## ACKNOWLEDGEMENTS

Huge thanks to Kwyn Meagher <https://blog.kwyn.io> for generously donating the package name in the main npm registry. Previously npx was used for a Tessel board Neopixels library, which can now be found under `npx-tessel` <https://npm.im/npx-tessel>.

## AUTHOR

Written by Kat Marchan <https://github.com/npm>.

## REPORTING BUGS

Please file any relevant issues on Github. <https://github.com/npm/npx>

## LICENSE

This work is released by its authors into the public domain under CC0-1.0. See LICENSE.md for details.

## SEE ALSO

? npm(1)

? npm-run-script(1)

? npm-config(7)

libnpv@10.2.1-2

March 2020

NPX(1)