

Force npm to use global installed node-gyp

Note: These instructions only work with npm 6 or older. For a solution that works with npm 8 (or older), see [Updating-npm-bundled-node-gyp.md](#).

Many issues are opened by users who are not running a current version of node-gyp.

npm bundles its own, internal, copy of node-gyp located at `npm/node_modules`, within npm's private dependencies which are separate from *globally* accessible packages. Therefore this internal copy of node-gyp is independent from any globally installed copy of node-gyp that may have been installed via `npm install -g node-gyp`.

So npm's internal copy of node-gyp **isn't** stored inside *global node_modules* and thus isn't available for use as a standalone package. npm uses it's *internal* copy of `node-gyp` to automatically build native addons.

When you install a *new* version of node-gyp outside of npm, it'll go into your *global node_modules*, but not under the `npm/node_modules` (where internal copy of node-gyp is stored). So it will get into your `$PATH` and you will be able to use this globally installed version (**but not internal node-gyp of npm**) as any other globally installed package.

The catch is that npm **won't** use global version unless you tell it to, it'll keep on using the **internal one**. You need to instruct it to by setting the `node_gyp` config variable (which goes into your `~/.npmrc`). You do this by running the `npm config set` command as below. Then npm will use the command in the path you supply whenever it needs to build a native addon.

Important: You also need to remember to unset this when you upgrade npm with a newer version of node-gyp, or you have to manually keep your globally installed node-gyp to date. See "Undo" below.

Linux and macOS

```
npm install --global node-gyp@latest
npm config set node_gyp $(npm prefix -g)/lib/node_modules/node-gyp/bin/node-gyp.js
```

sudo may be required for the first command if you get a permission error.

Windows

Windows Command Prompt

```
npm install --global node-gyp@latest
for /f "delims=" %P in ('npm prefix -g') do npm config set node_gyp "%P\node_modules\node-gyp\bin\node-gyp.js"
```

Powershell

```
npm install --global node-gyp@latest  
npm prefix -g | % {npm config set node_gyp "$_\node_modules\node-gyp\bin\node-gyp.js"}
```

Undo

Beware if you don't unset the `node_gyp` config option, npm will continue to use the globally installed version of node-gyp rather than the one it ships with, which may end up being newer.

```
npm config delete node_gyp  
npm uninstall --global node-gyp
```