

- v1.2.7 supports node 4+

Troubleshooting

- I'm getting EBADPLATFORM Unsupported platform for fsevents error.
- It's fine, nothing is broken. fsevents is macos-only. Other platforms are skipped. If you want to hide this warning, report a bug to NPM bugtracker asking them to hide ebadplatform warnings by default.

License

The MIT License Copyright (C) 2010-2020 by Philipp Dunkel, Ben Noordhuis, Elan Shankar, Paul Miller — see LICENSE file.

Visit our [GitHub page](#) and [NPM Page](#)

fsevents

Native access to MacOS FSEvents in [Node.js](#)

The FSEvents API in MacOS allows applications to register for notifications of changes to a given directory tree. It is a very fast and lightweight alternative to kqueue.

This is a low-level library. For a cross-platform file watching module that uses fsevents, check out [Chokidar](#).

Usage

```
npm install fsevents
```

Supports only **Node.js v8.16 and higher**.

```
const fsevents = require('fsevents');

// To start observation
const stop = fsevents.watch(__dirname,
  (path, flags, id) => {
    const info = fsevents.getInfo(path,
      flags);
  });

// To end observation
stop();
```

Important note: The API behaviour is slightly different from typical JS APIs. The stop function *must* be retrieved and stored somewhere, even if you don't plan to stop the watcher. If you forget it, the garbage collector will eventually kick in, the watcher will be unregistered, and your callbacks won't be called anymore.

The callback passed as the second parameter to .watch gets called whenever the operating system detects a change in the file system. It takes three arguments:

```
fsEvents.watch(dirname: string, path: string, flags: number, id: void) => Promise<undefined>
```

- path: string - the item in the filesystem that have been changed
 - flags: number - a numeric value describing what the change was
 - id: string - an unique-id identifying this specific event
- Returns closer callback which called returns a Promise resolving when the watcher process has been shut down.

```
fsEvents.getInfo(path: string, flags: number, id: string): fsEventInfo
```

The getInfo function takes the path, flags and id arguments and converts those parameters into a structure that is easier to digest to determine what the change was.

The fsEventsInfo has the following shape:

- v2.3 supports Apple Silicon ARM CPUs
- v2 supports node 8.16+ and reduces package size massively
- v1.2.8 supports node 6+

Changelog

```
/**
 * @typedef {
 *   {'created'|'modified'|'deleted'|'moved'|'changed'|'unknown'} fsEventsEv
 *   @typedef {
 *     {'file'|'directory'|'symlink'} fsEventsType
 *   }
 */
{
  "event": "created", // {fsEventsEvent}
  "path": "file.txt",
  "type": "file", // {fsEventsType}
  "changes": {
    "inode": true, // Had iNode Meta-
    "finder": false, // Had Finder Meta-
    "access": false, // Had access
    "permissions": false, // Had xAttributes
    "xattrs": false // Had xAttributes
  },
  "flags": 0x100000000
}
```