

- 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();
```

The `FSEvents` class has the following shape:

easier to digest to determine what the change was.

arguments and converts those parameters into a structure that is

The `getInfo` function takes the path, flags and id

`FSEvents.getInfo(path: string, flags: number, id: string): FSEventInfo`

resolving when the watcher process has been shut down.

Returns closer callback which when called returns a Promise

• `id: string` - an unique-id identifying this specific event

change was

• `flags: number` - a numeric value describing what the

changed

• `path: string` - the item in the filesystem that have been

changed

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

system. It takes three arguments:

The callback passed as the second parameter to `.watch` gets

any more.

unregistered, and your callbacks won't be called

collector will eventually kick in, the watcher will be

plan to stop the watcher. If you forget it, the garbage

be retrieved and stored somewhere, even if you don't

different from typical JS APIs. The `stop` function must

Important note: The API behaviour is slightly

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

Changelog

```

    {
      "tags": "0x100000000
    },
    changed
  "xattrs": false // Had XAttributes
  permissions changed
  "access": false // Had access
  Data changed
  "finders": false // Had Finder Meta-
  "inode": true // Had inode Meta-
  "inode": false // Had inode Meta-
  "changes": {}
  "type": "file", // {FSEventsType}
  "path": "file.txt",
  "event": "created", // {FSEventsEvent}
}

/*
 * @typedef FSEventsType
 * @typedef {file | directory | symlink} FSEventsEvent
 */

```