

Electron Fuses

Package time feature toggles

What are fuses?

For a subset of Electron functionality it makes sense to disable certain features for an entire application. For example, 99% of apps don't make use of `ELECTRON_RUN_AS_NODE`, these applications want to be able to ship a binary that is incapable of using that feature. We also don't want Electron consumers building Electron from source as that is both a massive technical challenge and has a high cost of both time and money.

Fuses are the solution to this problem, at a high level they are "magic bits" in the Electron binary that can be flipped when packaging your Electron app to enable / disable certain features / restrictions. Because they are flipped at package time before you code sign your app the OS becomes responsible for ensuring those bits aren't flipped back via OS level code signing validation (Gatekeeper / App Locker).

How do I flip the fuses?

The easy way

We've made a handy module, [@electron/fuses](#), to make flipping these fuses easy. Check out the README of that module for more details on usage and potential error cases.

```
require('@electron/fuses').flipFuses(  
  // Path to electron  
  require('electron'),  
  // Fuses to flip  
  {  
    runAsNode: false  
  }  
)
```

The hard way

Quick Glossary

- **Fuse Wire:** A sequence of bytes in the Electron binary used to control the fuses
- **Sentinel:** A static known sequence of bytes you can use to locate the fuse wire
- **Fuse Schema:** The format / allowed values for the fuse wire

Manually flipping fuses requires editing the Electron binary and modifying the fuse wire to be the sequence of bytes that represent the state of the fuses you want.

Somewhere in the Electron binary there will be a sequence of bytes that look like this:

```
| ...binary | sentinel_bytes | fuse_version | fuse_wire_length | fuse_wire |  
...binary |
```

- `sentinel_bytes` is always this exact string `dL7pKGdnNz796PbbjQWNKmHXBZaB9tsX`
- `fuse_version` is a single byte whose unsigned integer value represents the version of the fuse schema

- `fuse_wire_length` is a single byte whose unsigned integer value represents the number of fuses in the following fuse wire
- `fuse_wire` is a sequence of N bytes, each byte represents a single fuse and its state.
 - "0" (0x30) indicates the fuse is disabled
 - "1" (0x31) indicates the fuse is enabled
 - "r" (0x72) indicates the fuse has been removed and changing the byte to either 1 or 0 will have no effect.

To flip a fuse you find its position in the fuse wire and change it to "0" or "1" depending on the state you'd like.

You can view the current schema [here](#).