

Persistent Copyright & Licensing Information in Client-side JS, CSS &sim.

Let's make this work!

Matija Šuklje

2025-04-10

LLW 2025

what

what

... is the problem?

problem of client-side JS, CSS etc.

JS & CSS (&sim.) client-side code

- no notices at all, nor licenses
- even if in source, no notices when distributed

= copyright problem

first (controversial) proposal

first (controversial) proposal

goals

goals of the first proposal

just **how far** *can* we get with existing specs and tools? >;)=

- short
- simple
- no new specs – rely on existing standards
- no new tooling – rely on common tools

first (controversial) proposal

actual suggestion

actual suggestion: how it goes

- put copyright and license info into source code
- know what's what even when mixed with unmarked code
- (hopefully) have the info stay in place *even when minified*

[blog post of the original suggestion](#)

[FOSDEM workshop on this topic](#)

actual suggestion: how it looks

```
/*!  
 * SPDX-SnippetBegin  
 * SPDX-SnippetCopyrightText: © 2024 Hacke R. McRandom <hacker@mcrandom.example>  
 * SPDX-LicenseIdentifier: MIT OR Unlicense  
 */  
  
import half_of_npm  
  
code_goes_here();  
  
/*! SPDX-SnippetEnd */
```

does it work?

does it work?

in theory

support in minifiers

minifier	@license (JSDoc)	/*! (YUI)
Google Closure Compiler	yes	yes ¹
Esbuild	yes	yes
JShrink	yes	yes
SWC ²	yes	yes
Terser	yes	yes
UglifyJS	yes	yes
YUI Compressor	no	yes
Bun	no	no (but, wip)

¹It is not documented but [Google Closure Compiler](#) does indeed support ! and treats it explicitly as a license block (without any additional JSDoc parsing) [since 2016](#).

²SWC is primarily a compiler and module bundler, but I keep it here because it does its own minification instead of relying on an external tool.

does it work?

in practice

Svelte + Rollup + Terser

result: *mostly* fine

CSS:

- in `bundle.css` no SPDX tags, but OK since it was all 3rd party :)?

JS:

- in `bundle.js` all SPDX tags remain :)
- but one `SnippetEnd` tag came up before the `SnipppetBegin` tag :(
 - “problem” in how Terser associates comments to functions, fixable
- *but* in `bundle.js.map` tags are perfectly fine :)

PoC 2: React scripts

React Scripts (+ WebPack + Terser)

result: *mostly* fine – React = opinionated, v.hard to change

CSS:

- tags kept in both `main.05c219f8.css` and `main.05c219f8.css.map` :)

JS:

- `main.608edf8e.js` no SPDX tags (or any other comments) :(
- but `main.608edf8e.js.LICENSE.txt` has all the SPDX-* tags as well other important comments (e.g. @license blocks from 3rd party components) :)?
 - *all* license-relevant comments dumped into `main.608edf8e.js.LICENSE.txt` (useless)
- and `main.608edf8e.js.map` includes SPDX-* tags, as expected :)

Rollup + WebPack + Terser

we³ planted the SPDX tags in every *.js file during build

result: *very fine*

JS:

- in `rollup.browser.js` all SDPX tags seem to be in order; there is just some un-tagged code at the very bottom :)
- no tags in `rollup.browser.js.map`, but that was expected and correct

³It was actually Lukas Taegert-Atkinson, I merely presented the idea.

feedback so far

feedback so far

in broad strokes

compliance ppl: “yes, makes *total* sense”

compliance ppl: “yes, makes *total* sense”

JS devs: “no way, this will make the code way too big”

feedback so far

just how BIG does it get *really*?

theory

$$V_{min.wSPDX} = V_{min} + (n_{sourcefiles} \times V_{SPDXtagblock})$$

theory

$$V_{min.wSPDX} = V_{min} + (n_{sourcefiles} \times V_{SPDXtagblock})$$

- $V_{min.wSPDX}$... size of minified package with SPDX tags preserved
- V_{min} ... size of minified package
- $n_{sourcefiles}$... number of relevant source code files
- $V_{SPDXtagblock}$... size of SPDX tag block

theory to practice

should not be too much really

many small files – bigger impact

few big files – smaller impact

added size due to SPDX tags in minified code

practice

- PoC 1 (mini Svelte app): 12.324 B vs 12.681 B ($\Delta = +3\%$)
- PoC 2 (mini React app): 536 KiB vs 536 KiB ($\Delta = +0.2\%$)
- PoC 3 (Rollup Browser⁴): 374 KiB vs 415 KiB ($\Delta = +11\%$)

⁴NB: Rollup Browser PoC was the only one where even 3rd party files had added SPDX tags. In the others it was just 1st party code that we added SPDX tags to.

feedback so far

more specific feedback

use `REUSE.toml` instead and [banner with license info](#)

problem with this counter-proposal:

- solution depends on specific tool (Rollup)
- info is missing in source code – will get lost

big projects challenges from a big company

the proposal sounds like the proper way to go

but big project challenges:

- WebPack, UglifyJS etc. may over-preserve comments that are tagged as important, but aren't really (e.g. whole README)
 - performance hit; even worse with WebAssembly
 - problem is really in the misuse (or different interpretation) of what is “important” – potential argument to use `@license` instead of `/*!`

big projects challenges from a big company

the proposal sounds like the proper way to go

but big project challenges:

- WebPack, UglifyJS etc. may over-preserve comments that are tagged as important, but aren't really (e.g. whole README)
 - performance hit; even worse with WebAssembly
 - problem is really in the misuse (or different interpretation) of what is “important” – potential argument to use `@license` instead of `/*!`

current workaround applied by this company:

- create an SBOM relying on package manager data (and pretend data is good enough) and ship SBOM along minified JS

discussion

discussion

let's make a better proposal

make copyright & license data useful for compliance

that is

- short
- simple
- no new specs – rely on existing standards, *as far as possible*
- no new tooling – rely on common tools, *as far as possible*

discussion

some talking points

who cares about minified code?

do we even care about SPDX tags in *minified* code?

who cares about minified code?

do we even care about SPDX tags in *minified* code?

yes:

- it is useful info to have
- no need for new tooling

no:

- who cares – we clearly don't for other compiled code
- also, WASM / WebAssembly is potentially the future
- can we achieve the same with SourceMaps/banners/SBOM instead and link those via an `rel=license` attribute in either `<a>` or `<link>` HTML elements?
 - if so, we need to change some tools
 - but can potentially apply the REUSE spec on file-, instead of snippet-level

are SPDX IDs enough?

are SPDX tags (without license text) enough?

or are SPDX templates lacking legal clarity?

are SPDX IDs enough?

are SPDX tags (without license text) enough?

or are SPDX templates lacking legal clarity?

enough:

- SPDX ID = essentially URI for a (substantive) license text
- [B.3.4 Guideline: replaceable text](#) and [B.3.5 Guideline: omissible text](#) make sure that even with templated license texts these are substantively the same

we need more:

- strict verbatim reading of certain licenses

extension ideas

keep even more SPDX info – e.g. pURL or other origin ID

modify build tools to inject info:

- generate SPDX tags for 3rd party deps, if they did not have them
- ditto, but from a external source (LICENSE file or scanner)
- purl (Package URL)⁵
- ...

⁵Either as [External repository identifier](#): `SPDX-ExternalRef: PACKAGE-MANAGER purl $purl` or with the [Package download location field](#): `SPDX-PackageDownloadLocation: $purl`

fin

fin

thank you

matija@suklje.email

<https://matija.suklje.name/contact>