## Ke, Jessica - Intern

**From:** Friedman, Allan
**Sent:** Wednesday, June 16, 2021 8:49 PM
**To:** SBOM_RFC
**Subject:** Fw: Minimum Fields

---

**From:** JC Herz <jc.herz@ionchannel.io>
**Sent:** Wednesday, June 16, 2021 4:58 PM
**To:** Friedman, Allan <AFriedman@ntia.gov>
**Subject:** Minimum Fields

Allan -

By way of feedback to the NTIA call for feedback on minimum fields, it's important to understand whether an SBOM is produced pre-build, at-build-time or post-build. For SBOMs produced at-build-time or post-build, hashes are desirable - whether they should be required as minimum viable or not, at present, is a thorny business process and technical maturity question. A large number of software suppliers cannot meet that mark at present, and a higher bar may thwart adoption in the near-term, although the bar should definitely be raised as adoption increases. The pragmatic thing to do would be to drive adoption as broadly as possible, and then raise the bar. Any SBOM is miles better than no SBOM, for the simple reason that having to produce an SBOM drives unprecedented introspection by suppliers with minimal levels of maturity. The simple act of assembling or curating and inventory puts suppliers on another level of maturity - and it will be highly resisted if it's not seen as an attainable goal.

Regarding pre-build SBOMs produced from source code analysis, hashes of top-level projects ("this is the project being analyzed") are useful because they create auditability. Hashes for a project's direct or transitive dependencies are problematic because the software as a deployable entity doesn't actually exist - it hasn't been built. Between the time source code dependencies are specified and the time the software is built, the hashes of those dependencies - what's actually incorporated into the compiled software - may change. Today's pipeline tooling does not easily allow for build-time specification of particular hashes for dependencies, vs. dependency versions or tags. So the inclusion of hashes in a pre-build SBOM isn't necessarily an accurate representation of what's actually in the software, especially if that software is a binary package in a package manager. The actual dependencies at build-time may be dozens of commits ahead and have different hashes (and different risks). Package managers may even pull dependencies from different points of origin, based on proprietary algorithms native to the package managers. In that sense, pre-build hashes are chaff - they create ambiguity, which works against transparency.

In terms of the ability to understand and take action on a pre-build SBOM, i.e. for permission to use a source code component in a mission system, point of origin (git URL) for dependencies is more desirable and actionable than a dependency hash - although point of origin may be just as difficult to require from suppliers as a minimum field.

JC