An OMG® System Package Data Exchange Publication

System Package Data Exchange (SPDX)

Version 3.0

OMG Document Number	Formal/24-11-01
Release date	March 2025
Normative Reference:	https://www.omg.org/spec/SPDX

The System Package Data Exchange® (SPDX®) Specification Version 3.0

Use of Specification - Terms, Conditions & Notices

Trademarks

Compliance

Use of the SPDX trademarks is subject to the SPDX Trademark License, currently available at SPDX Legal Notices page.

Software developed under the terms of the licenses under which this specification is issued may claim compliance or conformance with this specification if and only if the software provider complies with the SPDX Trademark License given above.

Contents

Preface

OMG

Founded in 1989, the Object Management Group, Inc. (OMG) is an open membership, not-for-profit computer industry standards consortium that produces and maintains computer industry specifications for interoperable, portable, and reusable enterprise applications in distributed, heterogeneous environments. Membership includes Information Technology vendors, end users, government agencies, and academia.

OMG member companies write, adopt, and maintain its specifications following a mature, open process. OMG's specifications implement the Model Driven Architecture® (MDA®), maximizing ROI through a full-lifecycle approach to enterprise integration that covers multiple operating systems, programming languages, middleware and networking infrastructures, and software development environments. OMG's specifications include: UML® (Unified Modeling Language™); CORBA® (Common Object Request Broker Architecture); CWM™ (Common Warehouse Metamodel); and industry-specific standards for dozens of vertical markets.

More information on the OMG is available at https://www.omg.org/.

OMG Specifications

As noted, OMG specifications address middleware, modeling and vertical domain frameworks. All OMG Specifications are available from the OMG website at: https://www.omg.org/spec

All of OMG's formal specifications may be downloaded without charge from our website. (Products implementing OMG specifications are available from individual suppliers.) Copies of specifications, available in PostScript and PDF format, may be obtained from the Specifications Catalog cited above or by contacting the Object Management Group, Inc. at:

OMG Headquarters
9C Medway Road, PMB 274
Milford, MA 01757
USA
Tel: +1-781-444-0404
Fax: +1-781-444-0320
Email: pubs@omg.org

Certain OMG specifications are also available as ISO standards. Please consult https://www.iso.org

OMG's Issue Reporting Procedure

All OMG specifications are subject to continuous review and improvement. As part of this process we encourage readers to report any ambiguities, inconsistencies, or inaccuracies they may find by completing the Issue Reporting Form listed on the main web page https://www.omg.org, under Documents, Report a Bug/Issue.

1 Scope

The System Package Data Exchange (SPDX®) specification defines an open standard for communicating bill of materials (BOM) information for different topic areas.

SPDX defines an underlying data model as well as multiple serialization formats to encode that data model.

SPDX metadata includes details about creation and distribution, including the following:

- software composition, for collections of software (Packages), individual Files, and portions of files (Snippets)
- software build information
- artificial intelligence (AI) models
- datasets
- creator, supplier and distributor identity information
- provenance and integrity
- licenses and copyrights, including a curated list of licenses and exceptions
- security vulnerabilities, defects, and other quality data
- relationships between system elements
- software usage and lifecycle
- mechanisms to enable annotating SPDX elements and linking between multiple SPDX Documents

2 References

2.1 Normative References

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

Apache Maven, Apache Software Foundation, https://maven.apache.org/.

Bower API, https://bower.io/docs/api/#install.

Common Platform Enumeration (CPE) – Specification 2.2, The MITRE Corporation, https://cpe.mitre.org/files/cpe-specification_2.2.pdf.

Common Platform Enumeration (CPE): Naming Specification Version 2.3, NIST IR 7695, NIST, https://csrc.nist.gov/pubs/ir/7695/final.

Common Vulnerability Scoring System v3.0 (CVSS v3.0): Specification Document, Forum of Incident Response and Security Teams, Inc (FIRST), https://www.first.org/cvss/v3.0/specification-document.

Common Vulnerability Scoring System v3.1 (CVSS v3.1): Specification Document, Forum of Incident Response and Security Teams, Inc (FIRST), https://www.first.org/cvss/v3.1/specification-document.

Common Vulnerability Scoring System version 4.0 (CVSS v4.0): Specification Document, Forum of Incident Response and Security Teams, Inc (FIRST), https://www.first.org/cvss/v4.0/specification-document.

CVSS 3.0 schema, Forum of Incident Response and Security Teams, Inc (FIRST), https://www.first.org/cvss/cvss-v3.0.json.

CVSS 3.1 schema, Forum of Incident Response and Security Teams, Inc (FIRST), https://www.first.org/cvss/cvss-v3.1.json.

CVSS 4.0 schema, Forum of Incident Response and Security Teams, Inc (FIRST), https://www.first.org/cvss/cvss-v4.0.json.

EU general risk assessment methodology, European Commission, https://ec.europa.eu/docsroom/documents/17107.

npm-package.json, npm Inc., https://docs.npmjs.com/files/package.json.

NuGet documentation, Microsoft, https://docs.nuget.org/.

POSIX.1-2017 *The Open Group Base Specifications Issue 7*, 2018 edition, IEEE/Open Group, https://pubs.opengroup.org/onlinepubs/9699919799/.

Resource Description Framework (RDF), 2014-02-25, W3C, http://www.w3.org/standards/techs/rdf.

RFC 1319, *The MD2 Message-Digest Algorithm*, Internet Engineering Task Force, https://datatracker.ietf.org/doc/rfc1319/.

RFC 1320, *The MD4 Message-Digest Algorithm*, Internet Engineering Task Force, https://datatracker.ietf.org/doc/rfc1320/.

RFC 1321, *The MD5 Message-Digest Algorithm*, Internet Engineering Task Force, https://datatracker.ietf.org/doc/rfc1321/.

RFC 1950, *ZLIB Compressed Data Format Specification version 3.3*, Internet Engineering Task Force, https://datatracker.ietf.org/doc/rfc1950/.

RFC 2046, *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*, Internet Engineering Task Force, https://datatracker.ietf.org/doc/rfc2046/.

RFC 3174, *US Secure Hash Algorithm 1 (SHA1)*, Internet Engineering Task Force, https://datatracker.ietf.org/doc/rfc3174/.

RFC 3696, *Application Techniques for Checking and Transformation of Names*, Internet Engineering Task Force, https://datatracker.ietf.org/doc/rfc3696/.

RFC 3874, *A 224-bit One-way Hash Function: SHA-224*, Internet Engineering Task Force, https://datatracker.ietf.org/doc/rfc3874/.

RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*, Internet Engineering Task Force, https://datatracker.ietf.org/doc/rfc3986/.

RFC 5234, *Augmented BNF for Syntax Specifications: ABNF*, Internet Engineering Task Force, https://datatracker.ietf.org/doc/rfc5234/.

RFC 6234, *US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)*, Internet Engineering Task Force, https://datatracker.ietf.org/doc/rfc6234/.

RFC 7405, *Case-Sensitive String Support in ABNF*, Internet Engineering Task Force, https://datatracker.ietf.org/doc/rfc7405/.

RFC 7693, *The BLAKE2 Cryptographic Hash and Message Authentication Code (MAC)*, Internet Engineering Task Force, https://datatracker.ietf.org/doc/rfc7693/.

RFC 8259, *The JavaScript Object Notation (JSON) Data Interchange Format*, Internet Engineering Task Force, https://datatracker.ietf.org/doc/rfc8259/.

RFC 9393, *Concise Software Identification Tags*, Internet Engineering Task Force, https://datatracker.ietf.org/doc/rfc9393/.

Semantic Versioning 2.0.0, Tom Preston-Werner and SemVer contributors, https://semver.org.

SLSA Provenance v0.2, The Linux Foundation, https://slsa.dev/spec/v0.2/provenance.

SoftWare Heritage persistent IDentifiers (SWHIDs), in Draft International Standard *ISO/IEC DIS 18670 Information technology — SoftWare Hash IDentifier (SWHID) Specification V1.2*https://www.iso.org/standard/89985.html, also available at https://docs.softwareheritage.org/devel/swh-model/persistent-identifiers.html

SPDX and RDF Ontology, http://spdx.org/rdf/ontology/spdx-3-0-1

SPDX License List, The Linux Foundation, https://spdx.org/licenses/

SPDX License Exceptions, The Linux Foundation, https://spdx.org/licenses/exceptions-index.html

Stakeholder-Specific Vulnerability Categorization Guide, CISA, https://www.cisa.gov/stakeholder-specific-vulnerability-categorization-ssvc.

The EPSS Model, Forum of Incident Response and Security Teams, Inc (FIRST), https://www.first.org/epss/model.

Types of Software Bill of Material (SBOM) Documents, CISA, https://www.cisa.gov/sites/default/files/2023-04/sbom-types-document-508c.pdf.

2.2 Non-normative References

The following documents are referred to in the text.

1. CISQ Software Bill of Materials project, *Tool-to-Tool Software Bill of Materials Exchange*, https://www.it-cisq.org/software-bill-of-materials/
2. Dan Geer and Joshua Corman, *Almost Too Big to Fail*, Usenix ;login: article, Vol. 39. No. 4, August 2014, https://www.usenix.org/publications/login/august14/geer
3. Josh Corman, testimony at the Cybersecurity of the Internet of Things Hearing Before the Subcommittee on Information Technology of The Committee on Oversight and Government Reform House of Representatives One Hundred Fifteenth Congress First Session calling for software bill of materials in pending legislation, October 3, 2017, page 38, https://www.govinfo.gov/app/details/CHRG-115hhrg27760/CHRG-115hhrg27760

4. MITRE, *Standardizing SBOM within the SW Development Tooling Ecosystem*, Nov 2019, https://www.mitre.org/news-insights/publication/standardizing-sbom-within-sw-development-tooling-ecosystem
5. MITRE, *Deliver Uncompromised: Securing Critical Software Supply Chains Proposal to Establish an End-To-End Framework For Software Supply Chain Integrity*, Jan 2021, https://www.mitre.org/news-insights/publication/deliver-uncompromised-securing-critical-software-supply-chains
6. NTIA, *Notice of 07/19/18 Meeting of Multistakeholder Process on Promoting Software Component Transparency*, July 2018. https://www.ntia.gov/federal-register-notice/notice-071918-meeting-multistakeholder-process-promoting-software-component
7. NTIA Software Bill Of Materials web page, https://ntia.gov/sbom/
8. Open Source Initiative (OSI) Approved Licenses; https://opensource.org/licenses
9. Software Package Data Exchange (SPDX®) Specification Version 1.0 and 1.1, 1.2, 2.0, 2.1, 2.2 and 2.3; SPDX.dev, https://spdx.dev/specifications
10. The United States Department of Commerce, *The Minimum Elements For a Software Bill of Materials (SBOM) Pursuant to Executive Order 14028 on Improving the Nation's Cybersecurity*, Jul 2021, https://www.ntia.gov/report/2021/minimum-elements-software-bill-materials-sbom
11. White House, *Executive Order on Improving the Nation's Cybersecurity*, May 2021, https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/

3 Symbols

List of symbols/abbreviations.

3T-SBOM	Tool-to-Tool Software Bill of Materials Exchange
ABNF	Augmented Backus–Naur form
AI	Artificial Intelligence
BNF	Backus–Naur form
BOM	Bill of Materials
CISA	Cybersecurity and Infrastructure Security Agency
CISQ	Consortium for Information & Software Quality
CPE	Common Platform Enumeration
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
EPSS	Exploit Prediction Scoring System
ISO	International Organization for Standardization
JSON-LD	JavaScript Object Notation for Linking Data
KEV	Known Exploited Vulnerabilities
ML	Machine Learning
NIST	National Institute of Standards and Technology
NISTIR	NIST Internal or Interagency Report
NTIA	National Telecommunications and Information Administration
OSI	Open Source Initiative
OWL	Web Ontology Language
PAS	Publicly Available Specification
POSIX	Portable Operating System Interface
PTF	Platform Task Force
PURL	Package URL
RDF	Resource Description Framework
RFC	Request For Comments
SBOM	Software Bill of Materials
SHA	Secure Hash Algorithms
SHACL	Shapes Constraint Language
SPDX	System Package Data Exchange (previously Software Package Data Exchange)
SSVC	Stakeholder-Specific Vulnerability Categorization
SWHID	SoftWare Hash IDentifier
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VEX	Vulnerability Exploitability eXchange
XML	Extensible Markup Language

4 Terms and definitions

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at https://www.iso.org/obp
- IEC Electropedia: available at http://www.electropedia.org/

profile

A scope of usage for SPDX targeting support for particular use cases and scenarios (e.g., software, licensing, security, etc.). A profile identifies which particular SPDX namespaces, classes, and properties it leverages, along with any custom constraints unique to its use.

5 Conformance

5.1 Alternate notation for some conformance requirements

This standard contains more than a few cardinality assertions, each of which indicates the minimum and maximum number of times a property may appear. These are represented by using “minCount” and “maxCount” respectively. The absolute minimum number of occurrences is zero (0), while for an unbounded maximum number of occurrences a star (*) is being used.

Here are some examples:

- minCount: 1
- maxCount: *
- Cardinality: 0..1
- Cardinality: 0..*
- Cardinality: 1..1
- Cardinality: 1..*

Each of these assertions can easily be understood as to whether a feature is required, and if so, how many occurrences are required; also, whether a feature is permitted, and if so, in what number. As this is the format long familiar to the SPDX community, it has been preserved in this specification.

5.2 Introduction to Profiles

Profile is the term for a compliance point within the SPDX community across The Linux Foundation and OMG. The System Package Data Exchange (SPDX) specification defines the following nine compliance points, defined as “Profiles”:

- Core Profile
- Software Profile
- Security Profile
- Licensing Profile
- Dataset Profile
- AI Profile
- Build Profile
- Lite Profile
- Extension Profile

The Core Profile is mandatory. All others are optional.

5.3 Core Profile compliance point

The Core Profile includes the definitions of classes properties and vocabularies usable by all SPDX profiles when producing or consuming SPDX content. Although the classes, properties and vocabularies are somewhat extensive, the required fields are rather minimal to allow maximum flexibility while meeting minimum SBOM requirements. Software that conforms to the SPDX specification at the Core Profile compliance point shall be able to import and export serialized documents that conform with one of the defined SPDX serialization formats.

Conformance to the Core Profile compliance point is mandatory for all other SPDX profiles.

This compliance point, in combination with the Software Profile compliance point, provides a baseline of functionality that facilitates interchange of the bills of materials information produced by tools supporting SPDX.

5.4 Software Profile compliance point

The Software Profile includes the definitions of classes, properties and vocabularies for referring to and conveying information about software and is usable by all SPDX profiles when producing or consuming SPDX content.

Software that conforms to the SPDX specification at the Software Profile compliance point shall be able to import and export serialized documents that conform with one of the SPDX serialization formats defined SPDX serialization formats.

Conformance to the Software Profile compliance point does not entail support for the Licensing, Dataset, AI, Build, Lite, or Extension profiles of the SPDX.

This compliance point, in combination with the Core Profile compliance point, provides a baseline of functionality that facilitates interchange of the bills of materials information produced by tools supporting SPDX.

5.5 Security Profile compliance point

The Security Profile captures security-related information when producing or consuming SPDX content.

Software that conforms to the SPDX specification at the security profile compliance point shall be able to import and export serialized documents that conform with one of the SPDX serialization formats defined SPDX serialization formats, including the properties and relationships specified in the security profile, which are in support of exchanging information about software vulnerabilities that may exist, the severity of those vulnerabilities, and a mechanism to express how a vulnerability may affect a specific software element including if a fix is available.

Conformance to the Security Profile compliance point does not entail support for the Licensing, Dataset, AI, Build, Lite, or Extension profiles of the SPDX.

This compliance point facilitates interchange of the security information produced by tools supporting SPDX.

5.6 Licensing Profile compliance point

The Licensing Profile includes capturing details relevant to software licensing and intellectual property information when producing or consuming SPDX content. Specifically, software that conforms to the SPDX specification at the Licensing profile compliance point shall be able to import and export serialized documents that conform with one of the SPDX serialization formats defined SPDX serialization formats, including the classes and fields that comprise the SPDX License Expression syntax and that relate to the SPDX License List.

There are two associated profiles, the SimpleLicensing Profile and the ExpandedLicensing profiles. Both allow expression of the same information, albeit in different ways.

Conformance to the Licensing Profile compliance point does not entail support for the Software, Security, Dataset, AI, Build, Lite, or Extension profiles of the SPDX.

This compliance point facilitates interchange of the licensing documents expressing which licenses and copyright notices are determined by persons or automated tooling to apply to distributions of software that are produced by tools supporting SPDX.

5.7 Dataset Profile compliance point

The Dataset Profile captures the relevant information about the datasets used in an AI system or other applications when producing or consuming SPDX content.

Software that conforms to the SPDX specification at the Dataset Profile compliance point shall be able to import and export serialized documents that conform with one of the SPDX serialization formats defined SPDX serialization formats, including details such as dataset names, versions, sources, associated metadata, licensing information, and any other relevant attributes. The Dataset Profile can covey a description or summary of a dataset, including metadata, characteristics, and statistical information about the data. The Dataset Profile can convey insights into the structure, format, content, and properties of a dataset, helping users understand and analyze the data more effectively.

Conformance to the Dataset Profile compliance point does not entail support for the Software, Licensing, Security, AI, Build, Lite, or Extension profiles of the SPDX.

This compliance point facilitates interchange of the information about datasets produced by tools supporting SPDX.

5.8 AI Profile compliance point

The AI Profile captures an inventory list of software components and dependencies associated with an AI system when producing or consuming SPDX content.

Software that conforms to the SPDX specification at the AI Profile compliance point shall be able to import and export serialized documents that conform with one of the SPDX serialization formats defined SPDX serialization formats, including the information about software components and dependencies associated with artificial intelligence and machine learning (AI/ML) models and systems. This inventory includes the software frameworks, libraries, and other components used to build or deploy the AI system, along with relevant information about their versions, licenses, and useful security references including ethical and security information.

Conformance to the AI Profile compliance point does not entail support for the Software, Licensing, Security, Dataset, Build, Lite, or Extension profiles of the SPDX.

This compliance point facilitates interchange of the AI model related information produced by tools supporting SPDX.

5.9 Build Profile compliance point

The Build Profile captures build-related information when producing or consuming SPDX content.

Software that conforms to the SPDX specification at the Build Profile compliance point shall be able to import and export serialized documents that conform with one of the SPDX serialization formats defined SPDX serialization formats, including associated definitions to help express how software is generated and transformed. This includes encoding the inputs, outputs, procedures/instructions, environments and actors from the build process along with the associated evidence.

Conformance to the Build Profile compliance point does not entail support for the Software, Licensing, Security, Dataset, AI, Lite, or Extension profiles of the SPDX.

This compliance point facilitates interchange of the build information produced by tools supporting SPDX.

5.10 Lite Profile compliance point

The Lite Profile captures the minimum set of information required for license compliance in the software supply chain for producing or consuming SPDX content.

Software that conforms to the SPDX specification at the Lite Profile compliance point shall be able to import and export serialized documents that conform with one of the SPDX serialization formats defined SPDX serialization formats, including creation of the SBOM, package lists with licensing and other related items, and their relationships.

Conformance to the Lite Profile compliance point does not entail support for the Software, Licensing, Security, Dataset, AI, Build, or Extension profiles of the SPDX.

This compliance point facilitates interchange of minimal licensing information when produced by tools supporting SPDX.

5.11 Extension Profile compliance point

The Extension Profile captures extended tailored information when producing or consuming non-standard SPDX content in three ways:

- Support Profile-based extended characterization of Elements. Enables specification and expression of Element characterization extensions within any profile and namespace of SPDX without requiring changes to other profiles or namespaces and without requiring local subclassing of remote classes (which could inhibit ecosystem interoperability in some cases).
- Support extension of SPDX by adopting individuals or communities with Element characterization details uniquely specialized to their particular context. Enables adopting individuals or communities to utilize SPDX expressive capabilities along with expressing more arcane Element characterization details specific to them and not appropriate for standardization across SPDX.
- Support structured capture of expressive solutions for gaps in SPDX coverage from real-world use. Enables adopting individuals or communities to express Element characterization details they require that are not currently defined in SPDX but likely should be. Enables a practical pipeline that identifies gaps in SPDX that should be filled, expresses solutions to those gaps in a way that allows the identifying adopters to use the extended solutions with SPDX and does not conflict with current SPDX, can be clearly detected among the SPDX content exchange ecosystem, provides a clear and structured definition of gap solution that can be used as submission for revision to the SPDX standard.

Software that conforms to the SPDX specification at the Extension Profile compliance point shall be able to import and export serialized documents that conform with one of the SPDX serialization formats defined SPDX serialization formats, including the abstract Extension class serving as the base for all defined Extension subclasses.

Conformance to the Extension Profile compliance point does not entail support for the Licensing, Security, Dataset, AI, Build, or profiles of the SPDX but is expected to be used in combination with the other profiles to extend them.

This compliance point facilitates interchange of extended information that goes beyond the standard SPDX produced by tools supporting SPDX and is used between cooperating parties that understand the form of the extension and can produce and consume its non-standard content.

5.12 Trademark Compliance

To be designated an SPDX document, a file shall comply with the requirements of the SPDX Trademark License, as stated in the the SPDX Trademark Page.

The official copyright notice that shall be used with any verbatim reproduction and/or distribution of this SPDX Specification 3.0.1 is:

“Official SPDX® Specification 3.0.1 Copyright © 2010–2024 Linux Foundation and its Contributors. Licensed under the Community Specification License 1.0. All other rights are expressly reserved.”

The official copyright notice that shall be used with any non-verbatim reproduction and/or distribution of this SPDX Specification 3.0.1, including without limitation any partial use or combining this SPDX Specification with another work, is:

“This is not an official SPDX Specification. Portions herein have been reproduced from SPDX® Specification 3.0.1 found at spdx.dev. These portions are Copyright © 2010–2024 Linux Foundation and its Contributors, and are licensed under the Community Specification License 1.0 by the Linux Foundation and its Contributors. All other rights are expressly reserved by Linux Foundation and its Contributors.”

6 Model and serializations

6.1 Overview

This specification defines the data model of the SPDX standard, describing every piece of information about systems with software components. The data model is based on the Resource Description Framework (RDF) extensible knowledge representation data model, which provides a flexible and extensible way to represent and exchange information.

The data may be serialized in a variety of formats for storage and transmission.

6.2 RDF serialization

Since the data model is based on RDF, any SPDX data can be serialized in any of the multiple RDF serialization formats, including but not limited to:

- JSON-LD format as defined in JSON-LD 1.1;
- Turtle (Terse RDF Triple Language) format as defined in RDF 1.1 Turtle;
- N-Triples format as defined in RDF 1.1 N-Triples; and
- RDF/XML format as defined in RDF 1.1 XML Syntax.

The SPDX specification is accompanied by a JSON-LD context definition file that can be used to serialize SPDX in a much simpler and more human-readable JSON-LD format.

6.3 Canonical serialization

Canonical serialization is a single, consistent, normalized, deterministic, and reproducible form.

Such a canonical form normalizes things like ordering and formatting.

The content of the canonical serialization is exactly the same as the JSON-LD serialization of RDF data (see 4.2), just represented in a consistent way.

Canonical serialization is in JSON format, as defined in RFC 8259 (IETF STD 90), with the following additional characteristics:

- No line breaks
- Key names MUST be wrapped in double quotes
- No whitespace outside of strings
- `true`, `false` and `null`: the literal names must be lowercase; no other literal names are allowed

- Integers: represented in base 10 using decimal digits. This designates an integer component that may be prefixed with an optional minus sign. Leading zeros are not allowed.
- Strings: UTF-8 representation without specific canonicalisation. A string begins and ends with quotation marks (%x22). Any Unicode characters may be placed within the quotation marks, except for the two characters that MUST be escaped by a reverse solidus: quotation mark, reverse solidus, and the control characters (U+0000 through U+001F).
- Arrays: An array structure is represented as square brackets surrounding zero or more items. Items are separated by commas.
- Objects: An object structure is represented as a pair of curly brackets surrounding zero or more name/value pairs (or members). A name is a string containing only ASCII characters (0x21-0x7F). The names within an object must be unique. A single colon comes after each name, separating the name from the value. A single comma separates a value from a following name. The name/value pairs are ordered by name.

6.4 Serialization information

A collection of elements may be serialized in multiple formats.

An SpdxDocument element represents a collection of elements across all serialization data formats within the model.

The actual serialized bytes is represented by an Artifact element within the model.

A Relationship of type serializedInArtifact links an SpdxDocument to one or more serialized forms of itself.

When serializing a physical SpdxDocument, any property of the logical element that can be natively represented within the chosen serialization format (e.g., `@context` prefixes in JSON-LD instead of the namespaceMap) may utilize these native mechanisms. All remaining properties shall be serialized within the SpdxDocument element itself.

A serialization must not contain more than one SpdxDocument.

A given instance of serialization must not define more than one SpdxDocument element.

6.5 Serialization in JSON-LD

6.5.1 JSON-LD context file

JSON-LD contexts allow JSON documents to use simple, human-readable, locally defined terms while ensuring data interoperability across different systems.

The SPDX global JSON-LD context file must be used universally for all SPDX documents in JSON-LD format that adhere to a specific SPDX version.

SPDX global JSON-LD context file is available at: https://spdx.org/rdf/3.0.1/spdx-context.jsonld

All SPDX documents in JSON-LD format must include a reference to the SPDX global context file at the top level. This reference is achieved using the following JSON construct:

```
"@context": "https://spdx.org/rdf/3.0.1/spdx-context.jsonld"
```

The SPDX context file defines aliases for specific JSON-LD properties to improve compatibility with the SPDX model. These aliases are:

- `spdxId`: An alias for the `@id` property.
- `type`: An alias for the `@type` property.

Additional namespace mappings may be defined within a separate object within the context.

6.5.2 JSON-LD validation

An SPDX serialization in JSON-LD format is considered conformant to the SPDX specification if it adheres to the following two validation criteria:

- Structural validation: The JSON-LD document must structurally validate against the SPDX JSON Schema. This schema defines the expected structure of the JSON-LD document, including the required elements, data types, and permissible values.
- Semantic validation: The JSON-LD document must successfully validate against the SPDX OWL ontology. This ontology defines the expected relationships and constraints between SPDX elements. The SPDX OWL ontology also incorporates SHACL shape restrictions to further specify these constraints.

The SPDX JSON Schema is available at: https://spdx.org/schema/3.0.1/spdx-json-schema.json

The SPDX OWL ontology is available at: https://spdx.org/rdf/3.0.1/spdx-model.ttl

7 Additional information

7.1 Author acknowledgements

The following people authored this specification:

Adam Cohn, Adolfo García Veytia, Alan Tse, Alexios Zavras, Andrew Back, Ann Thornton, Armin Tänzer, Arthit Suriyawongkul, Ayumi Watanabe, Basil Peace, Bill Schineller, Bradlee Edmondson, Brandon Lum, Bruno Cornec, Ciaran Farrell, Daniel German, David Edelsohn, David Kemp, David A. Wheeler, Debra McGlade, Dennis Clark, Dick Brooks, Ed Warnicke, Eran Strod, Eric Thomas, Esteban Rockett, Gary O'Neall, Gopi Krishnan Rajbahadur, Guillaume Rousseau, Hassib Khanafer, Henk Birkholz, Hiroyuki Fukuchi, Itaru Hosomi, Jack Manbeck, Jaime Garcia, Jeff Licquia, Jeff Luszcz, Jeff Schutt, Jilayne Lovejoy, John Ellis, Jonas Oberg, Joshua Watt, Kamsang Salima, Karen Bennet, Karen Copenhaver, Kate Stewart, Kevin Mitchell, Kim Weins, Kirsten Newcomer, Kouki Hama, Kris Reeves, Liang Cao, Lon Hohberger, Marc-Etienne Vargenau, Mark Gisi, Marshall Clow, Martin Michlmayr, Martin von Willebrand, Mark Atwood, Matija Šuklje, Matt Germonprez, Maximilian Huber, Meret Behrens, Michael J. Herzog, Michel Ruffin, Nicole Pappler, Nisha Kumar, Nobuyuki Tanaka, Norio Kobota, Nuno Brito, Oliver Fendt, Paul Madick, Peter Williams, Phil Robb, Philip Koltun, Philip Odence, Philippe Ombredanne, Pierre Lapointe, Rana Rahal, Robert Martin, Robin Gandhi, Rose Judge, Sam Ellis, Sameer Ahmed, Satoru Koizumi, Scott K Peterson, Scott Lamons, Scott Sterling, Sean Barnum, Sebastian Crane, Shane Coughlan, Steve Cropper, Steve Winslow, Stuart Hughes, Takashi Ninjouji, Thomas F. Incorvia, Thomas Steenbergen, Tom Callaway, Tom Vidal, Toru Taima, Venkata Krishna, W. Trevor King, William Bartholomew, Yev Bronshteyn, Yoshiko Ouchi, Yoshiyuki Ito, Yuji Nomura, Yumi Tomita, and Zachary McFarland.

This page intentionally left blank.

8 Core

Summary

The basis for all SPDX profiles.

Description

The Core namespace defines foundational concepts serving as the basis for all SPDX-3.0 profiles.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core`

Name:	Core

8.1 Classes

8.1.1 Agent

Summary

Agent represents anything with the potential to act on a system.

Description

The Agent class represents anything that has the potential to act on a system.

This could be a person, organization, software agent, etc.

This is not to be confused with tools that are used to perform tasks.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/Agent`

Name:	Agent
Instantiability:	Concrete
SubclassOf:	Element

Superclasses

- /Core/Element

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
name	xsd:string	0	1
spdxId	xsd:anyURI	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

8.1.2 Annotation

Summary

An assertion made in relation to one or more elements.

Description

An Annotation is an assertion made in relation to one or more elements.

The `contentType` property describes the format of the `statement` property.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/Annotation`

Name:	Annotation
Instantiability:	Concrete
SubclassOf:	Element

Superclasses

- /Core/Element

Properties

Property	Type	minCount	maxCount
annotationType	AnnotationType	1	1
contentType	MediaType	0	1
statement	xsd:string	0	1
subject	Element	1	1

All properties (informative)

Property	Type	minCount	maxCount
annotationType	AnnotationType	1	1
comment	xsd:string	0	1
contentType	MediaType	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
name	xsd:string	0	1
spdxId	xsd:anyURI	1	1
statement	xsd:string	0	1
subject	Element	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

8.1.3 Artifact

Summary

A distinct article or unit within the digital domain.

Description

An artifact is a distinct article or unit within the digital domain, such as an electronic file, a software package, a device or an element of data.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/Artifact`

Name:	Artifact
Instantiability:	Abstract
SubclassOf:	Element

Superclasses

- /Core/Element

Properties

Property	Type	minCount	maxCount
builtTime	DateTime	0	1
originatedBy	Agent	0	*
releaseTime	DateTime	0	1
standardName	xsd:string	0	*
suppliedBy	Agent	0	1
supportLevel	SupportType	0	*
validUntilTime	DateTime	0	1

All properties (informative)

Property	Type	minCount	maxCount
builtTime	DateTime	0	1
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
name	xsd:string	0	1
originatedBy	Agent	0	*
releaseTime	DateTime	0	1
spdxId	xsd:anyURI	1	1
standardName	xsd:string	0	*
summary	xsd:string	0	1
suppliedBy	Agent	0	1
supportLevel	SupportType	0	*
validUntilTime	DateTime	0	1
verifiedUsing	IntegrityMethod	0	*

8.1.4 Bom

Summary

A container for a grouping of SPDX-3.0 content characterizing details (provenence, composition, licensing, etc.) about a product.

Description

A Bill of Materials (BOM) is a container for a grouping of SPDX-3.0 content characterizing details about a product.

This could include details of the content and composition of the product, provenence details of the product and/or its composition, licensing information, known quality or security issues, etc.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/Bom`

Name:	Bom
Instantiability:	Concrete
SubclassOf:	Bundle

Superclasses

- /Core/Bundle
- /Core/ElementCollection
- /Core/Element

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
context	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
element	Element	0	*
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
name	xsd:string	0	1
profileConformance	ProfileIdentifierType	0	*
rootElement	Element	0	*
spdxId	xsd:anyURI	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

8.1.5 Bundle

Summary

A collection of Elements that have a shared context.

Description

A bundle is a collection of Elements that have a shared context.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/Bundle`

Name:	Bundle
Instantiability:	Concrete
SubclassOf:	ElementCollection

Superclasses

- /Core/ElementCollection
- /Core/Element

Properties

Property	Type	minCount	maxCount
context	xsd:string	0	1

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
context	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
element	Element	0	*
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
name	xsd:string	0	1
profileConformance	ProfileIdentifierType	0	*
rootElement	Element	0	*
spdxId	xsd:anyURI	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

8.1.6 CreationInfo

Summary

Provides information about the creation of the Element.

Description

The CreationInfo provides information about who created the Element, and when and how it was created.

The dateTime created is often the date of last change (e.g., a git commit date), not the date when the SPDX data was created, as doing so supports reproducible builds.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/CreationInfo`

Name:	CreationInfo
Instantiability:	Concrete

Properties

Property	Type	minCount	maxCount
comment	xsd:string	0	1
created	DateTime	1	1
createdBy	Agent	1	*
createdUsing	Tool	0	*
specVersion	SemVer	1	1

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
created	DateTime	1	1
createdBy	Agent	1	*
createdUsing	Tool	0	*
specVersion	SemVer	1	1

8.1.7 DictionaryEntry

Summary

A key with an associated value.

Description

The class used for implementing a generic string mapping (also known as associative array, dictionary, or hash map) in SPDX.

Each DictionaryEntry contains a key-value pair which maps the key to its associated value.

To implement a dictionary, this class is to be used in a collection with unique keys.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/DictionaryEntry`

Name:	DictionaryEntry
Instantiability:	Concrete

Properties

Property	Type	minCount	maxCount
key	xsd:string	1	1
value	xsd:string	0	1

All properties (informative)

Property	Type	minCount	maxCount
key	xsd:string	1	1
value	xsd:string	0	1

8.1.8 Element

Summary

Base domain class from which all other SPDX-3.0 domain classes derive.

Description

An Element is a representation of a fundamental concept either directly inherent to the Bill of Materials (BOM) domain or indirectly related to the BOM domain and necessary for contextually characterizing BOM concepts and relationships. Within SPDX-3.0 structure this is the base class acting as a consistent, unifying, and interoperable foundation for all explicit and inter-relatable content objects.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/Element`

Name:	Element
Instantiability:	Abstract

Properties

Property	Type	minCount	maxCount
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
name	xsd:string	0	1
spdxId	xsd:anyURI	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
name	xsd:string	0	1
spdxId	xsd:anyURI	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

8.1.9 ElementCollection

Summary

A collection of Elements, not necessarily with unifying context.

Description

An ElementCollection is a collection of Elements, not necessarily with unifying context.

Note that all ElementCollections must conform to the Core profile even if the Core profile is not specified in the profileConformance property.

If the profileConformance property is not provided, “core” is to be assumed as the default.

Constraints

- If the ElementCollection has at least 1 element, it must also have at least 1 rootElement.
- The element must not be of type SpdxDocument.
- The rootElement must not be of type SpdxDocument.

Metadata

```
https://spdx.org/rdf/3.0.1/terms/Core/ElementCollection
```

Name:	ElementCollection
Instantiability:	Abstract
SubclassOf:	Element

Superclasses

- /Core/Element

Properties

Property	Type	minCount	maxCount
element	Element	0	*
profileConformance	ProfileIdentifierType	0	*
rootElement	Element	0	*

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
element	Element	0	*
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
name	xsd:string	0	1
profileConformance	ProfileIdentifierType	0	*
rootElement	Element	0	*
spdxId	xsd:anyURI	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

8.1.10 ExternalIdentifier

Summary

A reference to a resource identifier defined outside the scope of SPDX-3.0 content that uniquely identifies an Element.

Description

An ExternalIdentifier is a reference to a resource outside the scope of SPDX-3.0 content that provides a unique key within an established domain that can uniquely identify an Element.

Metadata

```
https://spdx.org/rdf/3.0.1/terms/Core/ExternalIdentifier
```

Name:	ExternalIdentifier
Instantiability:	Concrete

Properties

Property	Type	minCount	maxCount
comment	xsd:string	0	1
externalIdentifierType	ExternalIdentifierType	1	1
identifier	xsd:string	1	1
identifierLocator	xsd:anyURI	0	*
issuingAuthority	xsd:string	0	1

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
externalIdentifierType	ExternalIdentifierType	1	1
identifier	xsd:string	1	1
identifierLocator	xsd:anyURI	0	*
issuingAuthority	xsd:string	0	1

8.1.11 ExternalMap

Summary

A map of Element identifiers that are used within an SpdxDocument but defined external to that SpdxDocument.

Description

An external map is a map of Element identifiers that are used within an SpdxDocument but defined external to that SpdxDocument. The external map provides details about the externally-defined Element such as its provenance, where to retrieve it, and how to verify its integrity.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/ExternalMap`

Name:	ExternalMap
Instantiability:	Concrete

Properties

Property	Type	minCount	maxCount
definingArtifact	Artifact	0	1
externalSpdxId	xsd:anyURI	1	1
locationHint	xsd:anyURI	0	1
verifiedUsing	IntegrityMethod	0	*

All properties (informative)

Property	Type	minCount	maxCount
definingArtifact	Artifact	0	1
externalSpdxId	xsd:anyURI	1	1
locationHint	xsd:anyURI	0	1
verifiedUsing	IntegrityMethod	0	*

8.1.12 ExternalRef

Summary

A reference to a resource outside the scope of SPDX-3.0 content related to an Element.

Description

An External Reference points to a general resource outside the scope of the SPDX-3.0 content that provides additional context, characteristics or related information about an Element.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/ExternalRef`

Name:	ExternalRef
Instantiability:	Concrete

Properties

Property	Type	minCount	maxCount
comment	xsd:string	0	1
contentType	MediaType	0	1
externalRefType	ExternalRefType	0	1
locator	xsd:string	0	*

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
contentType	MediaType	0	1
externalRefType	ExternalRefType	0	1
locator	xsd:string	0	*

8.1.13 Hash

Summary

A mathematically calculated representation of a grouping of data.

Description

A hash is a grouping of characteristics unique to the result of applying a mathematical algorithm that maps data of arbitrary size to a bit string (the hash) and is a one-way function, that is, a function which is practically infeasible to invert.

This is commonly used for integrity checking of data.

Please note that different profiles may also provide additional methods for verifying the integrity of specific subclasses of Elements.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/Hash`

Name:	Hash
Instantiability:	Concrete
SubclassOf:	IntegrityMethod

Superclasses

- /Core/IntegrityMethod

Properties

Property	Type	minCount	maxCount
algorithm	HashAlgorithm	1	1
hashValue	xsd:string	1	1

All properties (informative)

Property	Type	minCount	maxCount
algorithm	HashAlgorithm	1	1
comment	xsd:string	0	1
hashValue	xsd:string	1	1

8.1.14 IndividualElement

Summary

A concrete subclass of Element used by Individuals in the Core profile.

Description

Individuals, such as NoneElement and NoAssertionElement, need to reference a concrete subclass of Element.

This class provides the type used by the individuals defined in the Core profile.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/IndividualElement`

Name:	IndividualElement
Instantiability:	Concrete
SubclassOf:	Element

Superclasses

- /Core/Element

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
name	xsd:string	0	1
spdxId	xsd:anyURI	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

8.1.15 IntegrityMethod

Summary

Provides an independently reproducible mechanism that permits verification of a specific Element.

Description

An IntegrityMethod provides an independently reproducible mechanism that permits verification of a specific Element that correlates to the data in this SPDX document. This identifier enables a recipient to determine if anything in the original Element has been changed and eliminates confusion over which version or modification of a specific Element is referenced.

Please note that different profiles may also provide additional methods for verifying the integrity of specific subclasses of Elements.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/IntegrityMethod`

Name:	IntegrityMethod
Instantiability:	Abstract

Properties

Property	Type	minCount	maxCount
comment	xsd:string	0	1

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1

8.1.16 LifecycleScopedRelationship

Summary

Provide context for a relationship that occurs in the lifecycle.

Description

Certain relationships are sensitive to where they occur in the lifecycle. This parameter lets us avoid a proliferation of relationships, by parameterizing this context information for a relationship.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/LifecycleScopedRelationship`

Name:	LifecycleScopedRelationship
Instantiability:	Concrete
SubclassOf:	Relationship

Superclasses

- /Core/Relationship
- /Core/Element

Properties

Property	Type	minCount	maxCount
scope	LifecycleScopeType	0	1

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
completeness	RelationshipCompleteness	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
endTime	DateTime	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
from	Element	1	1
name	xsd:string	0	1
relationshipType	RelationshipType	1	1
scope	LifecycleScopeType	0	1
spdxId	xsd:anyURI	1	1
startTime	DateTime	0	1
summary	xsd:string	0	1
to	Element	1	*
verifiedUsing	IntegrityMethod	0	*

8.1.17 NamespaceMap

Summary

A mapping between prefixes and namespace partial URIs.

Description

A namespace map allows the creator of a collection of serializable Elements to suggest shorter identifiers ("prefixes") for specific namespace portions of Element IDs. This map is used in SPDX content serialization to provide a more human-readable and smaller serialized representation of the Elements.

For details of how NamespaceMap content is to be serialized please refer to the Model and serializations[1] clause and the various serialization format-specific files within the spdx-3-model repository[2].

Namespace maps support a variety of relevant use cases such as:

1. An SPDX content producer wishing to provide clarity of their serialization of an SPDX 2.X simple style collection where all content is newly minted and a single prefix-namespace is used. The consumer of SPDX content wishes to preserve the name space mapping provided by such a producer.

   In this case, the consumer would record the namespace map prefixes in the NamespaceMap such that subsequent serializations could reproduce the prefixes / namespaces in the native serialization format.

[1] ../../../serializations.md

[2] https://github.com/spdx/spdx-3-model/tree/main/serialization

2. An SPDX content producer wishing to maintain consistent prefix use and understanding across multiple different serialization formats of the produced content.

   For example, an SBOM producer wishes to share/publish the SBOM as JSON-LD and XML. The producer can specify the preferred prefix mappings in the native serialization format using information from a single NamespaceMap accessible local to the producer.

3. An SPDX content consumer/producer wishing to maintain consistent prefix use while round tripping from SPDX content received, deserialized, modified/extended in some way, and then reserialized in the same serialization form.

   In this case the prefix-namespace mappings utilized in the content are transformed from the original native namespace/prefix into the in memory NamespaceMap then transformed from the NamespaceMap back into the resultant serialization native namespace / prefix format.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/NamespaceMap`

Name:	NamespaceMap
Instantiability:	Concrete

Properties

Property	Type	minCount	maxCount
namespace	xsd:anyURI	1	1
prefix	xsd:string	1	1

All properties (informative)

Property	Type	minCount	maxCount
namespace	xsd:anyURI	1	1
prefix	xsd:string	1	1

8.1.18 Organization

Summary

A group of people who work together in an organized way for a shared purpose.

Description

An Organization is a group of people who work together in an organized way for a shared purpose.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/Organization`

Name:	Organization
Instantiability:	Concrete
SubclassOf:	Agent

Superclasses

- /Core/Agent
- /Core/Element

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
name	xsd:string	0	1
spdxId	xsd:anyURI	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

8.1.19 PackageVerificationCode

Summary

An SPDX version 2.X compatible verification method for software packages.

Description

This verification method is provided for compatibility with SPDX 2.X.

Use of this verification code method is discouraged except for scenarios where the `contentIdentifier` property on `Artifact` can not be used.

This verification method provides an independently reproducible mechanism identifying specific contents of a package based on the actual files (except the SPDX document itself, if it is included in the package) that make up each package and that correlates to the data in this SPDX document.

This identifier enables a recipient to determine if any file in the original package (that the analysis was done on) has been changed and permits inclusion of an SPDX document as part of a package.

Algorithm:

```
templist = ""

for all files in the package {
    if file is a packageVerificationCodeExcludedFile
        skip it  /* exclude SPDX analysis file */
    else
        append "algorithm(file)/n" to templist
}

sort templist in ascending order by value

/* remove separators from ordered sequence */
valueslist = remove "/n"s from templist

if valueslist is empty
   hashValue = 0
else
   hashValue = algorithm(valueslist)
```

where `algorithm(string)` applies a hash algorithm on a string and returns the result in lowercase hexadecimal digits.

Required sort order: ‘0’, ‘1’, ‘2’, ‘3’, ‘4’, ‘5’, ‘6’, ‘7’, ‘8’, ‘9’, ‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’ (ASCII order)

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/PackageVerificationCode`

Name:	PackageVerificationCode
Instantiability:	Concrete
SubclassOf:	/Core/IntegrityMethod

Superclasses

- /Core/IntegrityMethod

Properties

Property	Type	minCount	maxCount
algorithm	HashAlgorithm	1	1
hashValue	xsd:string	1	1
packageVerificationCodeExcludedFile	xsd:string	0	*

All properties (informative)

Property	Type	minCount	maxCount
algorithm	HashAlgorithm	1	1
comment	xsd:string	0	1
hashValue	xsd:string	1	1
packageVerificationCodeExcludedFile	xsd:string	0	*

8.1.20 Person

Summary

An individual human being.

Description

A Person is an individual human being.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/Person`

Name:	Person
Instantiability:	Concrete
SubclassOf:	Agent

Superclasses

- /Core/Agent
- /Core/Element

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
name	xsd:string	0	1
spdxId	xsd:anyURI	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

8.1.21 PositiveIntegerRange

Summary

A tuple of two positive integers that define a range.

Description

PositiveIntegerRange is a tuple of two positive integers that define a range. “beginIntegerRange” must be less than or equal to “endIntegerRange”.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/PositiveIntegerRange`

Name:	PositiveIntegerRange
Instantiability:	Concrete

Properties

Property	Type	minCount	maxCount
beginIntegerRange	xsd:positiveInteger	1	1
endIntegerRange	xsd:positiveInteger	1	1

All properties (informative)

Property	Type	minCount	maxCount
beginIntegerRange	xsd:positiveInteger	1	1
endIntegerRange	xsd:positiveInteger	1	1

8.1.22 Relationship

Summary

Describes a relationship between one or more elements.

Description

A Relationship is a grouping of characteristics unique to an assertion that one Element is related to one or more other Elements in some way.

To explicitly assert that no such relationships exist, the `to` property should contain the NoneElement individual and no other elements.

A relationship that contains NoneElement and additional elements in the `to` property is not valid.

To explicitly assert that no assertions are being made regarding the existence of such relationships, the `to` property should contain the NoAssertionElement individual.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/Relationship`

Name:	Relationship
Instantiability:	Concrete
SubclassOf:	Element

Superclasses

- /Core/Element

Properties

Property	Type	minCount	maxCount
completeness	RelationshipCompleteness	0	1
endTime	DateTime	0	1
from	Element	1	1
relationshipType	RelationshipType	1	1
startTime	DateTime	0	1
to	Element	1	*

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
completeness	RelationshipCompleteness	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
endTime	DateTime	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
from	Element	1	1
name	xsd:string	0	1
relationshipType	RelationshipType	1	1
spdxId	xsd:anyURI	1	1
startTime	DateTime	0	1
summary	xsd:string	0	1
to	Element	1	*
verifiedUsing	IntegrityMethod	0	*

8.1.23 SoftwareAgent

Summary

A software agent.

Description

A SoftwareAgent is a software program that is given the authority (similar to a user's authority) to act on a system.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/SoftwareAgent`

Name:	SoftwareAgent
Instantiability:	Concrete
SubclassOf:	Agent

Superclasses

- /Core/Agent
- /Core/Element

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
name	xsd:string	0	1
spdxId	xsd:anyURI	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

8.1.24 SpdxDocument

Summary

A collection of SPDX Elements that could potentially be serialized.

Description

The SpdxDocument provides a convenient way to express information about collections of SPDX Elements that could potentially be serialized as complete units (e.g., all in-scope SPDX data within a single JSON-LD file).

SpdxDocument is independent of any particular serialization format or instance.

Information we wish to preserve about a specific instance of serialization of this SPDX content is NOT expressed using the SpdxDocument but rather using an associated Artifact representing a particular instance of SPDX data physical serialization.

Any instance of serialization of SPDX data MUST NOT contain more than one SpdxDocument element definition.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/SpdxDocument`

Name:	SpdxDocument
Instantiability:	Concrete
SubclassOf:	ElementCollection

Superclasses

- /Core/ElementCollection
- /Core/Element

Properties

Property	Type	minCount	maxCount
dataLicense	/SimpleLicensing/AnyLicenseInfo	0	1
import	ExternalMap	0	*
namespaceMap	NamespaceMap	0	*

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
dataLicense	/SimpleLicensing/AnyLicenseInfo	0	1
description	xsd:string	0	1
element	Element	0	*
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
import	ExternalMap	0	*
name	xsd:string	0	1
namespaceMap	NamespaceMap	0	*
profileConformance	ProfileIdentifierType	0	*
rootElement	Element	0	*
spdxId	xsd:anyURI	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

8.1.25 Tool

Summary

An element of hardware and/or software utilized to carry out a particular function.

Description

A Tool is an element of hardware and/or software utilized to carry out a particular function.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/Tool`

Name:	Tool
Instantiability:	Concrete
SubclassOf:	Element

Superclasses

- /Core/Element

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
name	xsd:string	0	1
spdxId	xsd:anyURI	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

8.2 Properties

8.2.1 algorithm

Summary

Specifies the algorithm used for calculating the hash value.

Description

An algorithm specifies the algorithm that was used for calculating the hash value.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/algorithm`

Name:	algorithm
Nature:	ObjectProperty
Range:	HashAlgorithm

Referenced

- /Core/Hash
- /Core/PackageVerificationCode

8.2.2 annotationType

Summary

Describes the type of annotation.

Description

An annotationType describes the type of an annotation.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/annotationType`

Name:	annotationType
Nature:	ObjectProperty
Range:	AnnotationType

Referenced

- /Core/Annotation

8.2.3 beginIntegerRange

Summary

Defines the beginning of a range.

Description

beginIntegerRange is a positive integer that defines the beginning of a range.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/beginIntegerRange`

Name:	beginIntegerRange
Nature:	DataProperty
Range:	xsd:positiveInteger

Referenced

- /Core/PositiveIntegerRange

8.2.4 builtTime

Summary

Specifies the time an artifact was built.

Description

A builtTime specifies the time an artifact was built.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/builtTime`

Name:	builtTime
Nature:	DataProperty
Range:	DateTime

Referenced

- /Core/Artifact

8.2.5 comment

Summary

Provide consumers with comments by the creator of the Element about the Element.

Description

A comment is an optional field for creators of the Element to provide comments to the readers/reviewers of the document.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/comment`

Name:	comment
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Core/CreationInfo
- /Core/Element
- /Core/ExternalIdentifier
- /Core/ExternalRef
- /Core/IntegrityMethod

8.2.6 completeness

Summary

Provides information about the completeness of relationships.

Description

Completeness gives information about whether the provided relationships are complete, known to be incomplete or if no assertion is made either way.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/completeness`

Name:	completeness
Nature:	ObjectProperty
Range:	RelationshipCompleteness

Referenced

- /Core/Relationship

8.2.7 contentType

Summary

Provides information about the content type of an Element or a Property.

Description

This field is a reasonable estimation of the content type of the Element or the Property, from a creator perspective.

Content type is intrinsic to the Element or the Property, independent of how it is being used.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/contentType`

Name:	contentType
Nature:	DataProperty
Range:	MediaType

Referenced

- /Core/Annotation
- /Core/ExternalRef
- /Software/File

8.2.8 context

Summary

Gives information about the circumstances or unifying properties that Elements of the bundle have been assembled under.

Description

A context gives information about the circumstances or unifying properties that Elements of the bundle have been assembled under.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/context`

Name:	context
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Core/Bundle

8.2.9 created

Summary

Identifies when the Element was originally created.

Description

Created is a date that identifies when the Element was originally created.

The time stamp can serve as an indication as to whether the analysis needs to be updated.

This is often the date of last change (e.g., a git commit date), not the date when the SPDX data was created, as doing so supports reproducible builds.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/created`

Name:	created
Nature:	DataProperty
Range:	DateTime

Referenced

- /Core/CreationInfo

8.2.10 createdBy

Summary

Identifies who or what created the Element.

Description

CreatedBy identifies who or what created the Element.

The generation method will assist the recipient of the Element in assessing the general reliability/accuracy of the analysis information.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/createdBy`

Name:	createdBy
Nature:	ObjectProperty
Range:	Agent

Referenced

- /Core/CreationInfo

8.2.11 createdUsing

Summary

Identifies the tooling that was used during the creation of the Element.

Description

CreatedUsing identifies the tooling that was used during the creation of the Element.

The generation method will assist the recipient of the Element in assessing the general reliability/accuracy of the analysis information.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/createdUsing`

Name:	createdUsing
Nature:	ObjectProperty
Range:	Tool

Referenced

- /Core/CreationInfo

8.2.12 creationInfo

Summary

Provides information about the creation of the Element.

Description

CreationInfo provides information about the creation of the Element.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/creationInfo`

Name:	creationInfo
Nature:	ObjectProperty
Range:	CreationInfo

Referenced

- /Core/Element

8.2.13 dataLicense

Summary

Provides the license under which the SPDX documentation of the Element can be used.

Description

The data license provides the license under which the SPDX documentation of the Element can be used.

This is to alleviate any concern that content (the data or database) in an SPDX file is subject to any form of intellectual property right that could restrict the re-use of the information or the creation of another SPDX file for the same project(s).

This approach avoids intellectual property and related restrictions over the SPDX file, however individuals can still contract with each other to restrict release of specific collections of SPDX files (which map to software bill of materials) and the identification of the supplier of SPDX files.

Compliance with this document includes populating the SPDX fields therein with data related to such fields (“SPDX-Metadata”).

This document contains numerous fields where an SPDX file creator may provide relevant explanatory text in SPDX-Metadata. Without opining on the lawfulness of “database rights” (in jurisdictions where applicable), such explanatory text is copyrightable subject matter in most Berne Convention countries.

By using the SPDX specification, or any portion hereof, you hereby agree that any copyright rights (as determined by your jurisdiction) in any SPDX-Metadata, including without limitation explanatory text, shall be subject to the terms of the Creative Commons CC0 1.0 Universal license.

For SPDX-Metadata not containing any copyright rights, you hereby agree and acknowledge that the SPDX-Metadata is provided to you “as-is” and without any representations or warranties of any kind concerning the SPDX-Metadata, express, implied, statutory or otherwise, including without limitation warranties of title, merchantability, fitness for a particular purpose, non-infringement, or the absence of latent or other defects, accuracy, or the presence or absence of errors, whether or not discoverable, all to the greatest extent permissible under applicable law.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/dataLicense`

Name:	dataLicense
Nature:	ObjectProperty
Range:	/SimpleLicensing/AnyLicenseInfo

Referenced

- /Core/SpdxDocument

8.2.14 definingArtifact

Summary

Artifact representing a serialization instance of SPDX data containing the definition of a particular Element.

Description

A definingArtifact property is used to link the Element identifier for an Element defined external to a given SpdxDocument to an Artifact Element representing the SPDX serialization instance which contains the definition for the Element.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/definingArtifact`

Name:	definingArtifact
Nature:	ObjectProperty
Range:	Artifact

Referenced

- /Core/ExternalMap

8.2.15 description

Summary

Provides a detailed description of the Element.

Description

This field is a detailed description of the Element. It may also be extracted from the Element itself.

The intent is to provide recipients of the SPDX file with a detailed technical explanation of the functionality, anticipated use, and anticipated implementation of the Element.

This field may also include a description of improvements over prior versions of the Element.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/description`

Name:	description
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Core/Element

8.2.16 element

Summary

Refers to one or more Elements that are part of an ElementCollection.

Description

This field refers to one or more Elements that are part of an ElementCollection.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/element`

Name:	element
Nature:	ObjectProperty
Range:	Element

Referenced

- /Core/ElementCollection

8.2.17 endIntegerRange

Summary

Defines the end of a range.

Description

endIntegerRange is a positive integer that defines the end of a range.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/endIntegerRange`

Name:	endIntegerRange
Nature:	DataProperty
Range:	xsd:positiveInteger

Referenced

- /Core/PositiveIntegerRange

8.2.18 endTime

Summary

Specifies the time from which an element is no longer applicable / valid.

Description

An endTime specifies the time from which element is no longer applicable / valid.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/endTime`

Name:	endTime
Nature:	DataProperty
Range:	DateTime

Referenced

- /Core/Relationship

8.2.19 extension

Summary

Specifies an Extension characterization of some aspect of an Element.

Description

`extension` specifies an Extension-based characterization of a particular aspect of an Element.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/extension`

Name:	extension
Nature:	ObjectProperty
Range:	/Extension/Extension

Referenced

- /Core/Element

8.2.20 externalIdentifier

Summary

Provides a reference to a resource outside the scope of SPDX-3.0 content that uniquely identifies an Element.

Description

ExternalIdentifier points to a resource outside the scope of SPDX-3.0 content that uniquely identifies an Element.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/externalIdentifier`

Name:	externalIdentifier
Nature:	ObjectProperty
Range:	ExternalIdentifier

Referenced

- /Core/Element

8.2.21 externalIdentifierType

Summary

Specifies the type of the external identifier.

Description

An externalIdentifierType specifies the type of the external identifier.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/externalIdentifierType`

Name:	externalIdentifierType
Nature:	ObjectProperty
Range:	ExternalIdentifierType

Referenced

- /Core/ExternalIdentifier

8.2.22 externalRef

Summary

Points to a resource outside the scope of the SPDX-3.0 content that provides additional characteristics of an Element.

Description

This field points to a resource outside the scope of the SPDX-3.0 content that provides additional characteristics of an Element.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/externalRef`

Name:	externalRef
Nature:	ObjectProperty
Range:	ExternalRef

Referenced

- /Core/Element

8.2.23 externalRefType

Summary

Specifies the type of the external reference.

Description

An externalRefType specifies the type of the external reference.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/externalRefType`

Name:	externalRefType
Nature:	ObjectProperty
Range:	ExternalRefType

Referenced

- /Core/ExternalRef

8.2.24 externalSpdxId

Summary

Identifies an external Element used within an SpdxDocument but defined external to that SpdxDocument.

Description

An externalSpdxId identifies an external Element used within an SpdxDocument but defined external to that SpdxDocument.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/externalSpdxId`

Name:	externalSpdxId
Nature:	DataProperty
Range:	xsd:anyURI

Referenced

- /Core/ExternalMap

8.2.25 from

Summary

References the Element on the left-hand side of a relationship.

Description

This field references the Element on the left-hand side of a relationship.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/from`

Name:	from
Nature:	ObjectProperty
Range:	Element

Referenced

- /Core/Relationship

8.2.26 hashValue

Summary

The result of applying a hash algorithm to an Element.

Description

A hashValue is the result of applying a hash algorithm to an Element.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/hashValue`

Name:	hashValue
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Core/Hash
- /Core/PackageVerificationCode

8.2.27 identifier

Summary

Uniquely identifies an external element.

Description

An identifier uniquely identifies an external element.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/identifier`

Name:	identifier
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Core/ExternalIdentifier

8.2.28 identifierLocator

Summary

Provides the location for more information regarding an external identifier.

Description

Identifiers are not always structured as URIs. An identifierLocator is a location hint (a URL) that provides contextual information relevant to the identifier.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/identifierLocator`

Name:	identifierLocator
Nature:	DataProperty
Range:	xsd:anyURI

Referenced

- /Core/ExternalIdentifier

8.2.29 import

Summary

Provides an ExternalMap of Element identifiers.

Description

An import provides an ExternalMap of an Element identifier that is used within a document but defined external to that document.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/import`

Name:	import
Nature:	ObjectProperty
Range:	ExternalMap

Referenced

- /Core/SpdxDocument

8.2.30 issuingAuthority

Summary

An entity that is authorized to issue identification credentials.

Description

An issuingAuthority is an entity that is authorized to issue identification credentials.

The entity may be a government, non-profit, educational institution, or commercial enterprise.

The string provides a unique identifier for the issuing authority.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/issuingAuthority`

Name:	issuingAuthority
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Core/ExternalIdentifier

8.2.31 key

Summary

A key used in a generic key-value pair.

Description

A key used in generic a key-value pair.

A key-value pair can be used to implement a dictionary which associates a key with a value.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/key`

Name:	key
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Core/DictionaryEntry

8.2.32 locationHint

Summary

Provides an indication of where to retrieve an external Element.

Description

A locationHint provides an indication of where to retrieve an external Element.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/locationHint`

Name:	locationHint
Nature:	DataProperty
Range:	xsd:anyURI

Referenced

- /Core/ExternalMap

8.2.33 locator

Summary

Provides the location of an external reference.

Description

A locator provides the location of an external reference.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/locator`

Name:	locator
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Core/ExternalRef

8.2.34 name

Summary

Identifies the name of an Element as designated by the creator.

Description

This field identifies the name of an Element as designated by the creator.

The name of an Element is an important convention and easier to refer to than the URI.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/name`

Name:	name
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Core/Element

8.2.35 namespace

Summary

Provides an unambiguous mechanism for conveying a URI fragment portion of an Element ID.

Description

A namespace provides an unambiguous mechanism for conveying a URI fragment portion of an Element ID.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/namespace`

Name:	namespace
Nature:	DataProperty
Range:	xsd:anyURI

Referenced

- /Core/NamespaceMap

8.2.36 namespaceMap

Summary

Provides a NamespaceMap of prefixes and associated namespace partial URIs applicable to an SpdxDocument and independent of any specific serialization format or instance.

Description

This field provides a NamespaceMap of prefixes and associated namespace partial URIs applicable to an SpdxDocument and independent of any specific serialization format or instance.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/namespaceMap`

Name:	namespaceMap
Nature:	ObjectProperty
Range:	NamespaceMap

Referenced

- /Core/SpdxDocument

8.2.37 originatedBy

Summary

Identifies from where or whom the Element originally came.

Description

An originatedBy identifies from where or whom the Element originally came.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/originatedBy`

Name:	originatedBy
Nature:	ObjectProperty
Range:	Agent

Referenced

- /Core/Artifact

8.2.38 packageVerificationCodeExcludedFile

Summary

The relative file name of a file to be excluded from the ‘PackageVerificationCode‘.

Description

A relative filename with the root of the package archive or directory referencing a file to be excluded from the `PackageVerificationCo`

Every filename is preceded with a `./`.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/packageVerificationCodeExcludedFile`

Name:	packageVerificationCodeExcludedFile
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Core/PackageVerificationCode

8.2.39 prefix

Summary

A substitute for a URI.

Description

A prefix is a substitute for a URI.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/prefix`

Name:	prefix
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Core/NamespaceMap

8.2.40 profileConformance

Summary

Describes one a profile which the creator of this ElementCollection intends to conform to.

Description

Describes a profile to which the creator of this ElementCollection intends to conform.

The profileConformance will apply to all Elements contained within the collection as well as the collection itself.

Conformance to a profile is defined by the additional restrictions documented in the profile specific documentation and schema files.

Use of this property allows the creator of an ElementCollection to communicate to consumers their intent to adhere to the profile additional restrictions.

The profileConformance has a default value of "core" if no other profileConformance is specified since all ElementCollections and Element must adhere to the Core profile.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/profileConformance`

Name:	profileConformance
Nature:	ObjectProperty
Range:	ProfileIdentifierType

Referenced

- /Core/ElementCollection

8.2.41 relationshipType

Summary

Information about the relationship between two Elements.

Description

This field provides information about the relationship between two Elements.

For example, you can represent a relationship between two different Files, between a Package and a File, between two Packages, or between one SpdxDocument and another SpdxDocument.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/relationshipType`

Name:	relationshipType
Nature:	ObjectProperty
Range:	RelationshipType

Referenced

- /Core/Relationship

8.2.42 releaseTime

Summary

Specifies the time an artifact was released.

Description

A releaseTime specifies the time an artifact was released.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/releaseTime`

Name:	releaseTime
Nature:	DataProperty
Range:	DateTime

Referenced

- /Core/Artifact

8.2.43 rootElement

Summary

This property is used to denote the root Element(s) of a tree of elements contained in a BOM.

Description

This property is used to denote the root Element(s) of a tree of elements contained in a BOM. The tree consists of other elements directly and indirectly related through properties or Relationships from the root.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/rootElement`

Name:	rootElement
Nature:	ObjectProperty
Range:	Element

Referenced

- /Core/ElementCollection

8.2.44 scope

Summary

Capture the scope of information about a specific relationship between elements.

Description

A scope is additional context about a relationship, that clarifies the relationship between elements.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/scope`

Name:	scope
Nature:	ObjectProperty
Range:	LifecycleScopeType

Referenced

- /Core/LifecycleScopedRelationship

8.2.45 spdxId

Summary

Identifies an Element to be referenced by other Elements.

Description

An spdxId uniquely identifies an Element which may thereby be referenced by other Elements. These references may be internal or external. While there may be several versions of the same Element, each one needs to be able to be referred to uniquely so that relationships between Elements can be clearly articulated.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/spdxId`

Name:	spdxId
Nature:	DataProperty
Range:	xsd:anyURI

Referenced

- /Core/Element

8.2.46 specVersion

Summary

Provides a reference number that can be used to understand how to parse and interpret an Element.

Description

The specVersion provides a reference number that can be used to understand how to parse and interpret an Element. It will enable both future changes to the specification and to support backward compatibility.

The major version number shall be incremented when incompatible changes between versions are made (one or more sections are created, modified or deleted). The minor version number shall be incremented when backwards compatible changes are made. The patch version number shall be incremented when backward compatible bug fixes are made.

Here, parties exchanging information in accordance with the SPDX specification need to provide 100% transparency as to which SPDX specification version such information is conforming to.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/specVersion`

Name:	specVersion
Nature:	DataProperty
Range:	SemVer

Referenced

- /Core/CreationInfo

8.2.47 standardName

Summary

The name of a relevant standard that may apply to an artifact.

Description

Various standards may be relevant or useful to capture for specific artifacts.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/standardName`

Name:	standardName
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Core/Artifact

8.2.48 startTime

Summary

Specifies the time from which an element is applicable / valid.

Description

A startTime specifies the time from which an element is applicable / valid.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/startTime`

Name:	startTime
Nature:	DataProperty
Range:	DateTime

Referenced

- /Core/Relationship

8.2.49 statement

Summary

Commentary on an assertion that an annotator has made.

Description

A statement is a commentary on an assertion that an annotator has made.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/statement`

Name:	statement
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Core/Annotation

8.2.50 subject

Summary

An Element an annotator has made an assertion about.

Description

A subject is an Element an annotator has made an assertion about.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/subject`

Name:	subject
Nature:	ObjectProperty
Range:	Element

Referenced

- /Core/Annotation

8.2.51 summary

Summary

A short description of an Element.

Description

A summary is a short description of an Element. Here, the intent is to allow the Element creator to provide concise information about the function or use of the Element.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/summary`

Name:	summary
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Core/Element

8.2.52 suppliedBy

Summary

Identifies who or what supplied the artifact or VulnAssessmentRelationship referenced by the Element.

Description

Identify the actual distribution source for the artifact (e.g., snippet, file, package, vulnerability) or VulnAssessmentRelationship being referenced.

This might or might not be different from the originating distribution source for the artifact (e.g., snippet, file, package, vulnerability) or VulnAssessmentRelationship.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/suppliedBy`

Name:	suppliedBy
Nature:	ObjectProperty
Range:	Agent

Referenced

- /Core/Artifact
- /Security/VulnAssessmentRelationship

8.2.53 supportLevel

Summary

Specifies the level of support associated with an artifact.

Description

supportLevel provides an indication of what support expectations that the supplier of an artifact is providing to the user.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/supportLevel`

Name:	supportLevel
Nature:	ObjectProperty
Range:	SupportType

Referenced

- /Core/Artifact

8.2.54 to

Summary

References an Element on the right-hand side of a relationship.

Description

This field references an Element on the right-hand side of a relationship.

If it is not provided, it indicates that there are no known relationships of the given type.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/to`

Name:	to
Nature:	ObjectProperty
Range:	Element

Referenced

- /Core/Relationship

8.2.55 validUntilTime

Summary

Specifies until when the artifact can be used before its usage needs to be reassessed.

Description

A validUntilTime specifies until when the artifact can be used before its usage needs to be reassessed.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/validUntilTime`

Name:	validUntilTime
Nature:	DataProperty
Range:	DateTime

Referenced

- /Core/Artifact

8.2.56 value

Summary

A value used in a generic key-value pair.

Description

A value used in a generic key-value pair.

A key-value pair can be used to implement a dictionary which associates a key with a value.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/value`

Name:	value
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Core/DictionaryEntry

8.2.57 verifiedUsing

Summary

Provides an IntegrityMethod with which the integrity of an Element can be asserted.

Description

A verifiedUsing provides an IntegrityMethod with which the integrity of an Element can be asserted.

Please note that different profiles may also provide additional methods for verifying the integrity of specific subclasses of Elements.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/verifiedUsing`

Name:	verifiedUsing
Nature:	ObjectProperty
Range:	IntegrityMethod

Referenced

- /Core/Element
- /Core/ExternalMap

8.3 Vocabularies

8.3.1 AnnotationType

Summary

Specifies the type of an annotation.

Description

AnnotationType specifies the type of an annotation.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/AnnotationType`

Name:	AnnotationType

Entries

other Used to store extra information about an Element which is not part of a review (e.g. extra information provided during the creation of the Element).

review Used when someone reviews the Element.

8.3.2 ExternalIdentifierType

Summary

Specifies the type of an external identifier.

Description

ExternalIdentifierType specifies the type of an external identifier.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/ExternalIdentifierType`

Name:	ExternalIdentifierType

Entries

cpe22 Common Platform Enumeration Specification 2.2[3]

cpe23 Common Platform Enumeration: Naming Specification Version 2.3[4]

cve Common Vulnerabilities and Exposures identifiers, an identifier for a specific software flaw defined within the official CVE Dictionary and that conforms to the CVE specification[5].

email Email address, as defined in RFC 3696[6] Section 3.

gitoid Gitoid[7], stands for Git Object ID[8]. A gitoid of type blob is a unique hash of a binary artifact. A gitoid may represent either an Artifact Identifier[9] for the software artifact or an Input Manifest Identifier[10] for the software artifact's associated Artifact Input Manifest[11]; this ambiguity exists because the Artifact Input Manifest is itself an artifact, and the gitoid of that artifact is its valid identifier. Gitoids calculated on software artifacts (Snippet, File, or Package Elements) should be recorded in the SPDX 3.0 SoftwareArtifact's contentIdentifier property. Gitoids calculated on the Artifact Input Manifest (Input Manifest Identifier) should be recorded in the SPDX 3.0 Element's externalIdentifier property. See OmniBOR Specification[12], a minimalistic specification for describing software Artifact Dependency Graphs[13].

other Used when the type does not match any of the other options.

packageUrl Package URL, as defined in the corresponding Annex[14] of this specification.

securityOther Used when there is a security related identifier of unspecified type.

swhid SoftWare Hash IDentifier, a persistent intrinsic identifier for digital artifacts, such as files, trees (also known as directories or folders), commits, and other objects typically found in version control systems. The format of the identifiers is defined in the SWHID specification[15] (ISO/IEC DIS 18670). They typically look like `swh:1:cnt:94a9ed024d3859793618152ea559a168`

swid Concise Software Identification (CoSWID) tag, as defined in RFC 9393[16] Section 2.3.

urlScheme Uniform Resource Identifier (URI) Schemes[17]. The scheme used in order to locate a resource.

8.3.3 ExternalRefType

Summary

Specifies the type of an external reference.

Description

ExternalRefType specifies the type of an external reference.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/ExternalRefType`

Name:	ExternalRefType

[3] https://cpe.mitre.org/files/cpe-specification_2.2.pdf

[4] https://csrc.nist.gov/publications/detail/nistir/7695/final

[5] https://csrc.nist.gov/glossary/term/cve_id

[6] https://datatracker.ietf.org/doc/rfc3986/

[7] https://www.iana.org/assignments/uri-schemes/prov/gitoid

[8] https://git-scm.com/book/en/v2/Git-Internals-Git-Objects

[9] https://github.com/omnibor/spec/blob/eb1ee5c961c16215eb8709b2975d193a2007a35d/spec/SPEC.md#artifact-identifier-types

[10] https://github.com/omnibor/spec/blob/eb1ee5c961c16215eb8709b2975d193a2007a35d/spec/SPEC.md#input-manifest-identifier

[11] https://github.com/omnibor/spec/blob/eb1ee5c961c16215eb8709b2975d193a2007a35d/spec/SPEC.md#artifact-input-manifest

[12] https://github.com/omnibor/spec/

[13] https://github.com/omnibor/spec/blob/eb1ee5c961c16215eb8709b2975d193a2007a35d/spec/SPEC.md#artifact-dependency-graph-adg

[14] ../../../annexes/pkg-url-specification.md

[15] https://www.swhid.org/specification/v1.1/4.Syntax

[16] https://datatracker.ietf.org/doc/rfc9393/

[17] https://www.iana.org/assignments/uri-schemes/uri-schemes.xhtml

Entries

altDownloadLocation A reference to an alternative download location.

altWebPage A reference to an alternative web page.

binaryArtifact A reference to binary artifacts related to a package.

bower A reference to a Bower package. The package locator format, looks like `package#version`, is defined in the "install" section of Bower API documentation[18].

buildMeta A reference build metadata related to a published package.

buildSystem A reference build system used to create or publish the package.

certificationReport A reference to a certification report for a package from an accredited/independent body.

chat A reference to the instant messaging system used by the maintainer for a package.

componentAnalysisReport A reference to a Software Composition Analysis (SCA) report.

cwe Common Weakness Enumeration[19]. A reference to a source of software flaw defined within the official CWE List[20] that conforms to the CWE specification[21].

documentation A reference to the documentation for a package.

dynamicAnalysisReport A reference to a dynamic analysis report for a package.

eolNotice A reference to the End Of Sale (EOS) and/or End Of Life (EOL) information related to a package.

exportControlAssessment A reference to a export control assessment for a package.

funding A reference to funding information related to a package.

issueTracker A reference to the issue tracker for a package.

license A reference to additional license information related to an artifact.

mailingList A reference to the mailing list used by the maintainer for a package.

mavenCentral A reference to a Maven repository artifact. The artifact locator format is defined in the Maven documentation[22] and looks like `groupId:artifactId[:version]`.

metrics A reference to metrics related to package such as OpenSSF scorecards.

npm A reference to an npm package. The package locator format is defined in the npm documentation[23] and looks like `package@version`.

nuget A reference to a NuGet package. The package locator format is defined in the NuGet documentation[24] and looks like `package/version`.

other Used when the type does not match any of the other options.

privacyAssessment A reference to a privacy assessment for a package.

productMetadata A reference to additional product metadata such as reference within organization's product catalog.

purchaseOrder A reference to a purchase order for a package.

qualityAssessmentReport A reference to a quality assessment for a package.

releaseHistory A reference to a published list of releases for a package.

releaseNotes A reference to the release notes for a package.

[18] https://bower.io/docs/api/#install

[19] https://csrc.nist.gov/glossary/term/common_weakness_enumeration

[20] https://cwe.mitre.org/data/

[21] https://cwe.mitre.org/

[22] https://maven.apache.org/guides/mini/guide-naming-conventions.html

[23] https://docs.npmjs.com/cli/v10/configuring-npm/package-json

[24] https://docs.nuget.org

riskAssessment A reference to a risk assessment for a package.

runtimeAnalysisReport A reference to a runtime analysis report for a package.

secureSoftwareAttestation A reference to information assuring that the software is developed using security practices as defined by NIST SP 800-218 Secure Software Development Framework (SSDF) Version 1.1[25] or CISA Secure Software Development Attestation Form[26].

securityAdversaryModel A reference to the security adversary model for a package.

securityAdvisory A reference to a published security advisory (where advisory as defined per ISO 29147:2018[27]) that may affect one or more elements, e.g., vendor advisories or specific NVD entries.

securityFix A reference to the patch or source code that fixes a vulnerability.

securityOther A reference to related security information of unspecified type.

securityPenTestReport A reference to a penetration test[28] report for a package.

securityPolicy A reference to instructions for reporting newly discovered security vulnerabilities for a package.

securityThreatModel A reference the security threat model[29] for a package.

socialMedia A reference to a social media channel for a package.

sourceArtifact A reference to an artifact containing the sources for a package.

staticAnalysisReport A reference to a static analysis report for a package.

support A reference to the software support channel or other support information for a package.

vcs A reference to a version control system related to a software artifact.

vulnerabilityDisclosureReport A reference to a Vulnerability Disclosure Report (VDR) which provides the software supplier's analysis and findings describing the impact (or lack of impact) that reported vulnerabilities have on packages or products in the supplier's SBOM as defined in NIST SP 800-161 Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations[30].

vulnerabilityExploitabilityAssessment A reference to a Vulnerability Exploitability eXchange (VEX) statement which provides information on whether a product is impacted by a specific vulnerability in an included package and, if affected, whether there are actions recommended to remediate. See also NTIA VEX one-page summary[31].

8.3.4 HashAlgorithm

Summary

A mathematical algorithm that maps data of arbitrary size to a bit string.

Description

A HashAlgorithm is a mathematical algorithm that maps data of arbitrary size to a bit string (the hash) and is a one-way function, that is, a function which is practically infeasible to invert.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/HashAlgorithm`

Name:	HashAlgorithm

[25] https://csrc.nist.gov/pubs/sp/800/218/final

[26] https://www.cisa.gov/resources-tools/resources/secure-software-development-attestation-form

[27] https://www.iso.org/standard/72311.html

[28] https://en.wikipedia.org/wiki/Penetration_test

[29] https://en.wikipedia.org/wiki/Threat_model

[30] https://csrc.nist.gov/pubs/sp/800/161/r1/final

[31] https://ntia.gov/files/ntia/publications/vex_one-page_summary.pdf

Entries

adler32 Adler-32 checksum is part of the widely used zlib compression library as defined in RFC 1950[32] Section 2.3.

blake2b256 BLAKE2b algorithm with a digest size of 256, as defined in RFC 7693[33] Section 4.

blake2b384 BLAKE2b algorithm with a digest size of 384, as defined in RFC 7693[34] Section 4.

blake2b512 BLAKE2b algorithm with a digest size of 512, as defined in RFC 7693[35] Section 4.

blake3 BLAKE3[36]

crystalsDilithium Dilithium[37]

crystalsKyber Kyber[38]

falcon FALCON[39]

md2 MD2 message-digest algorithm, as defined in RFC 1319[40].

md4 MD4 message-digest algorithm, as defined in RFC 1186[41].

md5 MD5 message-digest algorithm, as defined in RFC 1321[42].

md6 MD6 hash function[43]

other any hashing algorithm that does not exist in this list of entries

sha1 SHA-1, a secure hashing algorithm, as defined in RFC 3174[44].

sha224 SHA-2 with a digest length of 224, as defined in RFC 3874[45].

sha256 SHA-2 with a digest length of 256, as defined in RFC 6234[46].

sha384 SHA-2 with a digest length of 384, as defined in RFC 6234[47].

sha3_224 SHA-3 with a digest length of 224, as defined in FIPS 202[48].

sha3_256 SHA-3 with a digest length of 256, as defined in FIPS 202[49].

sha3_384 SHA-3 with a digest length of 384, as defined in FIPS 202[50].

sha3_512 SHA-3 with a digest length of 512, as defined in FIPS 202[51].

sha512 SHA-2 with a digest length of 512, as defined in RFC 6234[52].

8.3.5 LifecycleScopeType

Summary

Provide an enumerated set of lifecycle phases that can provide context to relationships.

[32] https://datatracker.ietf.org/doc/rfc1950/
[33] https://datatracker.ietf.org/doc/rfc7693/
[34] https://datatracker.ietf.org/doc/rfc7693/
[35] https://datatracker.ietf.org/doc/rfc7693/
[36] https://github.com/BLAKE3-team/BLAKE3-specs/blob/master/blake3.pdf
[37] https://pq-crystals.org/dilithium/
[38] https://pq-crystals.org/kyber/
[39] https://falcon-sign.info/falcon.pdf
[40] https://datatracker.ietf.org/doc/rfc1319/
[41] https://datatracker.ietf.org/doc/rfc1186/
[42] https://datatracker.ietf.org/doc/rfc1321/
[43] https://people.csail.mit.edu/rivest/pubs/RABCx08.pdf
[44] https://datatracker.ietf.org/doc/rfc3174/
[45] https://datatracker.ietf.org/doc/rfc3874/
[46] https://datatracker.ietf.org/doc/rfc6234/
[47] https://datatracker.ietf.org/doc/rfc6234/
[48] https://csrc.nist.gov/pubs/fips/202/final
[49] https://csrc.nist.gov/pubs/fips/202/final
[50] https://csrc.nist.gov/pubs/fips/202/final
[51] https://csrc.nist.gov/pubs/fips/202/final
[52] https://datatracker.ietf.org/doc/rfc6234/

Description

This enumeration summarizes common phases when dependency and other relationships, have different implications, based on their context. For example, a build dependency, may have different implications than a operational dependency.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/LifecycleScopeType`

Name:	LifecycleScopeType

Entries

build A relationship has specific context implications during an element's build phase, during development.

design A relationship has specific context implications during an element's design.

development A relationship has specific context implications during development phase of an element.

other A relationship has other specific context information necessary to capture that the above set of enumerations does not handle.

runtime A relationship has specific context implications during the execution phase of an element.

test A relationship has specific context implications during an element's testing phase, during development.

8.3.6 PresenceType

Summary

Categories of presence or absence.

Description

This type is used to indicate if a given field is present or absent or unknown.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/PresenceType`

Name:	PresenceType

Entries

no Indicates absence of the field.

noAssertion Makes no assertion about the field.

yes Indicates presence of the field.

8.3.7 ProfileIdentifierType

Summary

Enumeration of the valid profiles.

Description

There are a set of profiles that have been defined by a profile team.

A profile consists of a namespace that may add properties and classes to the Core profile unique to the domain covered by the profile.

The profile may also contain additional restrictions on existing properties and classes defined in other profiles.

If the creator of an SPDX collection of elements includes a profile in the list of profileConformance, they are claiming that all contained elements conform to all restrictions defined for that profile.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/ProfileIdentifierType`

Name:	ProfileIdentifierType

Entries

ai the element follows the AI profile specification

build the element follows the Build profile specification

core the element follows the Core profile specification

dataset the element follows the Dataset profile specification

expandedLicensing the element follows the ExpandedLicensing profile specification

extension the element follows the Extension profile specification

lite the element follows the Lite profile specification

security the element follows the Security profile specification

simpleLicensing the element follows the SimpleLicensing profile specification

software the element follows the Software profile specification

8.3.8 RelationshipCompleteness

Summary

Indicates whether a relationship is known to be complete, incomplete, or if no assertion is made with respect to relationship completeness.

Description

RelationshipCompleteness indicates whether the provided relationship is known to be complete, known to be incomplete, or if no assertion is made by the relationship creator.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/RelationshipCompleteness`

Name:	RelationshipCompleteness

Entries

complete The relationship is known to be exhaustive.

incomplete The relationship is known not to be exhaustive.

noAssertion No assertion can be made about the completeness of the relationship.

8.3.9 RelationshipType

Summary

Information about the relationship between two Elements.

Description

Provides information about the relationship between two Elements. For example, you can represent a relationship between two different Files, between a Package and a File, between two Packages, or between one SpdxDocument and another SpdxDocument.

Relationship names should be descriptive enough to easily deduce the correct direction from their name. The best way to do this is to make sure that the relationship name completes the sentence:

`from` (is) (a) `RELATIONSHIP to`

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/RelationshipType`

Name:	RelationshipType

Entries

affects The from Vulnerability affects each to Element. The use of the affects type is constrained to VexAffectedVulnAssessmen classed relationships.

amendedBy The from Element is amended by each to Element.

ancestorOf The from Element is an ancestor of each to Element.

availableFrom The from Element is available from the additional supplier described by each to Element.

configures The from Element is a configuration applied to each to Element, during a LifecycleScopeType period.

contains The from Element contains each to Element.

coordinatedBy The from Vulnerability is coordinatedBy the to Agent(s) (vendor, researcher, or consumer agent).

copiedTo The from Element has been copied to each to Element.

delegatedTo The from Agent is delegating an action to the Agent of the to Relationship (which must be of type invokedBy), during a LifecycleScopeType (e.g. the to invokedBy Relationship is being done on behalf of from).

dependsOn The from Element depends on each to Element, during a LifecycleScopeType period.

descendantOf The from Element is a descendant of each to Element.

describes The from Element describes each to Element. To denote the root(s) of a tree of elements in a collection, the rootElement property should be used.

doesNotAffect The from Vulnerability has no impact on each to Element. The use of the doesNotAffect is constrained to VexNotAffectedVulnAssessmentRelationship classed relationships.

expandsTo The from archive expands out as an artifact described by each to Element.

exploitCreatedBy The from Vulnerability has had an exploit created against it by each to Agent.

fixedBy Designates a from Vulnerability has been fixed by the to Agent(s).

fixedIn A from Vulnerability has been fixed in each to Element. The use of the fixedIn type is constrained to VexFixedVulnAssessmentRelationship classed relationships.

foundBy Designates a from Vulnerability was originally discovered by the to Agent(s).

generates The from Element generates each to Element.

hasAddedFile Every to Element is a file added to the from Element (from hasAddedFile to).

hasAssessmentFor Relates a from Vulnerability and each to Element with a security assessment. To be used with VulnAssessmentRelationship types.

hasAssociatedVulnerability Used to associate a from Artifact with each to Vulnerability.

hasConcludedLicense The from SoftwareArtifact is concluded by the SPDX data creator to be governed by each to license.

hasDataFile The from Element treats each to Element as a data file. A data file is an artifact that stores data required or optional for the from Element's functionality. A data file can be a database file, an index file, a log file, an AI model file, a calibration data file, a temporary file, a backup file, and more. For AI training dataset, test dataset, test artifact, configuration data, build input data, and build output data, please consider using the more specific relationship types: trainedOn, testedOn, hasTest, configures, hasInput, and hasOutput, respectively. This relationship does not imply dependency.

hasDeclaredLicense The from SoftwareArtifact was discovered to actually contain each to license, for example as detected by use of automated tooling.

hasDeletedFile Every to Element is a file deleted from the from Element (from hasDeletedFile to).

hasDependencyManifest The from Element has manifest files that contain dependency information in each to Element.

hasDistributionArtifact The from Element is distributed as an artifact in each to Element (e.g. an RPM or archive file).

hasDocumentation The from Element is documented by each to Element.

hasDynamicLink The `from` Element dynamically links in each `to` Element, during a LifecycleScopeType period.

hasEvidence Every `to` Element is considered as evidence for the `from` Element (`from` hasEvidence `to`).

hasExample Every `to` Element is an example for the `from` Element (`from` hasExample `to`).

hasHost The `from` Build was run on the `to` Element during a LifecycleScopeType period (e.g. the host that the build runs on).

hasInput The `from` Build has each `to` Element as an input, during a LifecycleScopeType period.

hasMetadata Every `to` Element is metadata about the `from` Element (`from` hasMetadata `to`).

hasOptionalComponent Every `to` Element is an optional component of the `from` Element (`from` hasOptionalComponent `to`).

hasOptionalDependency The `from` Element optionally depends on each `to` Element, during a LifecycleScopeType period.

hasOutput The `from` Build element generates each `to` Element as an output, during a LifecycleScopeType period.

hasPrerequisite The `from` Element has a prerequisite on each `to` Element, during a LifecycleScopeType period.

hasProvidedDependency The `from` Element has a dependency on each `to` Element, dependency is not in the distributed artifact, but assumed to be provided, during a LifecycleScopeType period.

hasRequirement The `from` Element has a requirement on each `to` Element, during a LifecycleScopeType period.

hasSpecification Every `to` Element is a specification for the `from` Element (`from` hasSpecification `to`), during a LifecycleScopeType period.

hasStaticLink The `from` Element statically links in each `to` Element, during a LifecycleScopeType period.

hasTest Every `to` Element is a test artifact for the `from` Element (`from` hasTest `to`), during a LifecycleScopeType period.

hasTestCase Every `to` Element is a test case for the `from` Element (`from` hasTestCase `to`).

hasVariant Every `to` Element is a variant the `from` Element (`from` hasVariant `to`).

invokedBy The `from` Element was invoked by the `to` Agent, during a LifecycleScopeType period (for example, a Build element that describes a build step).

modifiedBy The `from` Element is modified by each `to` Element.

other Every `to` Element is related to the `from` Element where the relationship type is not described by any of the SPDX relationship types (this relationship is directionless).

packagedBy Every `to` Element is a packaged instance of the `from` Element (`from` packagedBy `to`).

patchedBy Every `to` Element is a patch for the `from` Element (`from` patchedBy `to`).

publishedBy Designates a `from` Vulnerability was made available for public use or reference by each `to` Agent.

reportedBy Designates a `from` Vulnerability was first reported to a project, vendor, or tracking database for formal identification by each `to` Agent.

republishedBy Designates a `from` Vulnerability's details were tracked, aggregated, and/or enriched to improve context (i.e. NVD) by each `to` Agent.

serializedInArtifact The `from` SpdxDocument can be found in a serialized form in each `to` Artifact.

testedOn The `from` Element has been tested on the `to` Element(s).

trainedOn The `from` Element has been trained on the `to` Element(s).

underInvestigationFor The `from` Vulnerability impact is being investigated for each `to` Element. The use of the `underInvestigationFor` type is constrained to `VexUnderInvestigationVulnAssessmentRelationship` classed relationships.

usesTool The `from` Element uses each `to` Element as a tool, during a LifecycleScopeType period.

8.3.10 SupportType

Summary

Indicates the type of support that is associated with an artifact.

Description

SupportType is an enumeration of the various types of support commonly found for artifacts in the software supply chain. Specific details of what that support entails are provided by agreements between the producer and consumer of the artifact.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/SupportType`

Name:	SupportType

Entries

deployed in addition to being supported by the supplier, the software is known to have been deployed and is in use. For a software as a service provider, this implies the software is now available as a service.

development the artifact is in active development and is not considered ready for formal support from the supplier.

endOfSupport there is a defined end of support for the artifact from the supplier. This may also be referred to as end of life. There is a validUntilDate that can be used to signal when support ends for the artifact.

limitedSupport the artifact has been released, and there is limited support available from the supplier. There is a validUntilDate that can provide additional information about the duration of support.

noAssertion no assertion about the type of support is made. This is considered the default if no other support type is used.

noSupport there is no support for the artifact from the supplier, consumer assumes any support obligations.

support the artifact has been released, and is supported from the supplier. There is a validUntilDate that can provide additional information about the duration of support.

8.4 Individuals

8.4.1 NoAssertionElement

Summary

An Individual Value for Element representing a set of Elements of unknown identify or cardinality (number).

Description

NoAssertionElement should be used if

- the SPDX creator has attempted to but cannot reach a reasonable objective determination;
- the SPDX creator has made no attempt to determine this field; or
- the SPDX creator has intentionally provided no information (no meaning should be implied by doing so).

For example, a Relationship with `relationshipType`="ancestorOf", `from`=Element1, and `to`=NoAssertionElement is explicitly expressing that no assertion is being made about any potential descendants of Element1.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/NoAssertionElement`

Name:	NoAssertionElement
Type:	IndividualElement
IRI:	`https://spdx.org/rdf/3.0.1/terms/Core/NoAssertionElement`

8.4.2 NoneElement

Summary

An Individual Value for Element representing a set of Elements with cardinality (number/count) of zero.

Description

NoneElement should be used if the SPDX creator desires to assert that there are NO elements for the given context of use.

For example, a Relationship with `relationshipType`="ancestorOf", `from`=Element1, and `to`=NoneElement is explicitly expressing an assertion that Element1 has no descendants.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/NoneElement`

Name:	NoneElement
Type:	IndividualElement
IRI:	`https://spdx.org/rdf/3.0.1/terms/Core/NoneElement`

8.4.3 SpdxOrganization

Summary

An Organization representing the SPDX Project.

Description

SpdxOrganization is an Organization representing the SPDX Project. It is by definition the creator of all Element type individuals defined by the SPDX Project. These individuals include licences and exceptions defined in the SPDX License List, as well as individuals defined in the specification.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/SpdxOrganization`

Name:	SpdxOrganization
Type:	Organization
IRI:	`https://spdx.org/`

8.5 Datatypes

8.5.1 DateTime

Summary

A string representing a specific date and time.

Description

A DateTime is a string representation of a specific date and time.

It has resolution of seconds and is always expressed in UTC time zone.

The specific format is one of the most commonly used ISO-8601 formats.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/DateTime`

Name:	DateTime
SubclassOf:	xsd:dateTimeStamp

Format pattern

```
^\d\d\d\d-\d\d-\d\dT\d\d:\d\d:\d\dZ$
```

8.5.2 MediaType

Summary

Standardized way of indicating the type of content of an Element or a Property. A String constrained to the RFC 2046 specification.

Description

A MediaType is a string constrained to the RFC 2046 MIME Part Two: Media Types[53]. It provides a standardized way of indicating the type of content of an Element or a Property.

Example

- `application/java-archive`

[53] https://datatracker.ietf.org/doc/rfc2046/

- `application/vcard+json`
- `application/vnd.oasis.opendocument.text`
- `image/avif`
- `text/csv;charset=UTF-8`
- `text/javascript`
- `text/spdx`

A list of all possible media types is available at IANA Protocol Registries[54].

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/MediaType`

Name:	MediaType
SubclassOf:	xsd:string

Format pattern

```
^[^\/]+\/[^\/]+$
```

8.5.3 SemVer

Summary

A string constrained to the SemVer 2.0.0 specification.

Description

A semantic version is a string that is following the specification of Semantic Versioning 2.0.0[55].

Metadata

`https://spdx.org/rdf/3.0.1/terms/Core/SemVer`

Name:	SemVer
SubclassOf:	xsd:string

Format pattern

```
^(0|[1-9]\d*)\.(0|[1-9]\d*)\.(0|[1-9]\d*)(?:-((?:0|[1-9]\d*|\d*[a-zA-Z-][0-9a-zA-Z-]*)(?:\.(
```

[54] https://www.iana.org/assignments/media-types/media-types.xhtml

[55] https://semver.org/

This page intentionally left blank.

9 Software

Summary

Everything having to do with software.

Description

The Software namespace defines concepts related to software artifacts.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software`

Name:	Software

9.1 Classes

9.1.1 ContentIdentifier

Summary

A canonical, unique, immutable identifier

Description

A ContentIdentifier is a canonical, unique, immutable identifier of the content of a software artifact, such as a package, a file, or a snippet.

It can be used for verifying its identity and integrity.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/ContentIdentifier`

Name:	ContentIdentifier
Instantiability:	Concrete
SubclassOf:	/Core/IntegrityMethod

Superclasses

- /Core/IntegrityMethod

Properties

Property	Type	minCount	maxCount
contentIdentifierType	ContentIdentifierType	1	1
contentIdentifierValue	xsd:anyURI	1	1

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
contentIdentifierType	ContentIdentifierType	1	1
contentIdentifierValue	xsd:anyURI	1	1

9.1.2 File

Summary

Refers to any object that stores content on a computer.

Description

Refers to any object that stores content on a computer. The type of content can optionally be provided in the contentType property.

The fileKind property can be set to `directory` to indicate the file represents a directory and all content stored in that directory.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/File`

Name:	File
Instantiability:	Concrete
SubclassOf:	/Software/SoftwareArtifact

Superclasses

- /Software/SoftwareArtifact
- /Core/Artifact
- /Core/Element

Properties

Property	Type	minCount	maxCount
/Core/contentType	/Core/MediaType	0	1
fileKind	FileKindType	0	1

External properties cardinality updates

Property	minCount	maxCount
/Core/Element/name	1	

All properties (informative)

Property	Type	minCount	maxCount
additionalPurpose	SoftwarePurpose	0	*
attributionText	xsd:string	0	*
builtTime	DateTime	0	1
comment	xsd:string	0	1
contentIdentifier	ContentIdentifier	0	*
contentType	/Core/MediaType	0	1
copyrightText	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
fileKind	FileKindType	0	1
name	xsd:string	1	1
originatedBy	Agent	0	*
primaryPurpose	SoftwarePurpose	0	1
releaseTime	DateTime	0	1
spdxId	xsd:anyURI	1	1
standardName	xsd:string	0	*
summary	xsd:string	0	1
suppliedBy	Agent	0	1
supportLevel	SupportType	0	*
validUntilTime	DateTime	0	1
verifiedUsing	IntegrityMethod	0	*

9.1.3 Package

Summary

Refers to any unit of content that can be associated with a distribution of software.

Description

A package refers to any unit of content that can be associated with a distribution of software.

Typically, a package is composed of one or more files.

Any of the following non-limiting examples may be (but are not required to be) represented in SPDX as a package:

- a tarball, zip file or other archive
- a directory or sub-directory
- a separately distributed piece of software which another Package or File uses or depends upon (e.g., a Python package, a Go module, …)
- a container image, and/or each image layer within a container image
- a collection of one or more sub-packages
- a Git repository snapshot from a particular point in time

Note that some of these could be represented in SPDX as a file as well.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/Package`

Name:	Package
Instantiability:	Concrete
SubclassOf:	/Software/SoftwareArtifact

Superclasses

- /Software/SoftwareArtifact
- /Core/Artifact
- /Core/Element

Properties

Property	Type	minCount	maxCount
downloadLocation	xsd:anyURI	0	1
homePage	xsd:anyURI	0	1
packageUrl	xsd:anyURI	0	1
packageVersion	xsd:string	0	1
sourceInfo	xsd:string	0	1

External properties cardinality updates

Property	minCount	maxCount
/Core/Element/name	1	

All properties (informative)

Property	Type	minCount	maxCount
additionalPurpose	SoftwarePurpose	0	*
attributionText	xsd:string	0	*
builtTime	DateTime	0	1
comment	xsd:string	0	1
contentIdentifier	ContentIdentifier	0	*
copyrightText	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
downloadLocation	xsd:anyURI	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
homePage	xsd:anyURI	0	1
name	xsd:string	1	1
originatedBy	Agent	0	*
packageUrl	xsd:anyURI	0	1
packageVersion	xsd:string	0	1
primaryPurpose	SoftwarePurpose	0	1
releaseTime	DateTime	0	1
sourceInfo	xsd:string	0	1
spdxId	xsd:anyURI	1	1
standardName	xsd:string	0	*
summary	xsd:string	0	1
suppliedBy	Agent	0	1
supportLevel	SupportType	0	*
validUntilTime	DateTime	0	1
verifiedUsing	IntegrityMethod	0	*

9.1.4 Sbom

Summary

A collection of SPDX Elements describing a single package.

Description

A Software Bill of Materials (SBOM) is a collection of SPDX Elements describing a single package.

This could include details of the content and composition of the product, provenance details of the product and/or its composition, licensing information, known quality or security issues, etc.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/Sbom`

Name:	Sbom
Instantiability:	Concrete
SubclassOf:	/Core/Bom

Superclasses

- /Core/Bom
- /Core/Bundle
- /Core/ElementCollection
- /Core/Element

Properties

Property	Type	minCount	maxCount
sbomType	SbomType	0	*

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
context	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
element	Element	0	*
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
name	xsd:string	0	1
profileConformance	ProfileIdentifierType	0	*
rootElement	Element	0	*
sbomType	SbomType	0	*
spdxId	xsd:anyURI	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

9.1.5 Snippet

Summary

Describes a certain part of a file.

Description

A Snippet describes a certain part of a file and can be used when the file is known to have some content that has been included from another original source.

Snippets are useful for denoting when part of a file may have been originally created under another license or copied from a place with a known vulnerability.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/Snippet`

Name:	Snippet
Instantiability:	Concrete
SubclassOf:	/Software/SoftwareArtifact

Superclasses

- /Software/SoftwareArtifact
- /Core/Artifact
- /Core/Element

Properties

Property	Type	minCount	maxCount
byteRange	/Core/PositiveIntegerRange	0	1
lineRange	/Core/PositiveIntegerRange	0	1
snippetFromFile	File	1	1

All properties (informative)

Property	Type	minCount	maxCount
additionalPurpose	SoftwarePurpose	0	*
attributionText	xsd:string	0	*
builtTime	DateTime	0	1
byteRange	/Core/PositiveIntegerRange	0	1
comment	xsd:string	0	1
contentIdentifier	ContentIdentifier	0	*
copyrightText	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
lineRange	/Core/PositiveIntegerRange	0	1
name	xsd:string	0	1
originatedBy	Agent	0	*
primaryPurpose	SoftwarePurpose	0	1
releaseTime	DateTime	0	1
snippetFromFile	File	1	1
spdxId	xsd:anyURI	1	1
standardName	xsd:string	0	*
summary	xsd:string	0	1
suppliedBy	Agent	0	1
supportLevel	SupportType	0	*
validUntilTime	DateTime	0	1
verifiedUsing	IntegrityMethod	0	*

9.1.6 SoftwareArtifact

Summary

A distinct article or unit related to Software.

Description

A software artifact is a distinct article or unit related to software such as a package, a file, or a snippet.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/SoftwareArtifact`

Name:	SoftwareArtifact
Instantiability:	Abstract
SubclassOf:	/Core/Artifact

Superclasses

- /Core/Artifact
- /Core/Element

Properties

Property	Type	minCount	maxCount
additionalPurpose	SoftwarePurpose	0	*
attributionText	xsd:string	0	*
contentIdentifier	ContentIdentifier	0	*
copyrightText	xsd:string	0	1
primaryPurpose	SoftwarePurpose	0	1

All properties (informative)

Property	Type	minCount	maxCount
additionalPurpose	SoftwarePurpose	0	*
attributionText	xsd:string	0	*
builtTime	DateTime	0	1
comment	xsd:string	0	1
contentIdentifier	ContentIdentifier	0	*
copyrightText	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
name	xsd:string	0	1
originatedBy	Agent	0	*
primaryPurpose	SoftwarePurpose	0	1
releaseTime	DateTime	0	1
spdxId	xsd:anyURI	1	1
standardName	xsd:string	0	*
summary	xsd:string	0	1
suppliedBy	Agent	0	1
supportLevel	SupportType	0	*
validUntilTime	DateTime	0	1
verifiedUsing	IntegrityMethod	0	*

9.2 Properties

9.2.1 additionalPurpose

Summary

Provides additional purpose information of the software artifact.

Description

An additionalPurpose provides information about the additional purpose of the software artifact in addition to the primaryPurpose.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/additionalPurpose`

Name:	additionalPurpose
Nature:	ObjectProperty
Range:	SoftwarePurpose

Referenced

- /Software/SoftwareArtifact

9.2.2 attributionText

Summary

Provides a place for the SPDX data creator to record acknowledgement text for a software Package, File or Snippet.

Description

An attributionText for a software Package, File or Snippet provides a consumer of SPDX data with acknowledgement content, to assist redistributors of the Package, File or Snippet with reproducing those acknowledgements.

For example, this field may include a statement that is required by a particular license to be reproduced in end-user documentation, advertising materials, or another form.

This field may describe where, or in which contexts, the acknowledgements need to be reproduced, but it is not required to do so. The SPDX data creator may also explain elsewhere (such as in a comment field) how they intend for data in this field to be used.

An attributionText is not meant to include the software Package, File or Snippet's actual complete license text. Use hasConcludedLicense to identify the corresponding license.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/attributionText`

Name:	attributionText
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Software/SoftwareArtifact

9.2.3 byteRange

Summary

Defines the byte range in the original host file that the snippet information applies to.

Description

This field defines the byte range in the original host file that the snippet information applies to.

A range of bytes is independent of various formatting concerns, and the most accurate way of referring to the differences. The choice was made to start the numbering of the byte range at 1 to be consistent with the W3C pointer method vocabulary.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/byteRange`

Name:	byteRange
Nature:	DataProperty
Range:	/Core/PositiveIntegerRange

Referenced

- /Software/Snippet

9.2.4 contentIdentifier

Summary

A canonical, unique, immutable identifier of the artifact content, that may be used for verifying its identity and/or integrity.

Description

A contentIdentifier is a canonical, unique, immutable identifier of the content of a software artifact, such as a package, a file, or a snippet.

It may be used for verifying its identity and/or integrity.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/contentIdentifier`

Name:	contentIdentifier
Nature:	DataProperty
Range:	ContentIdentifier

Referenced

- /Software/SoftwareArtifact

9.2.5 contentIdentifierType

Summary

Specifies the type of the content identifier.

Description

A contentIdentifierType specifies the type of the content identifier.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/contentIdentifierType`

Name:	contentIdentifierType
Nature:	ObjectProperty
Range:	ContentIdentifierType

Referenced

- /Software/ContentIdentifier

9.2.6 contentIdentifierValue

Summary

Specifies the value of the content identifier.

Description

A contentIdentifierValue specifies the value of a content identifier.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/contentIdentifierValue`

Name:	contentIdentifierValue
Nature:	DataProperty
Range:	xsd:anyURI

Referenced

- /Software/ContentIdentifier

9.2.7 copyrightText

Summary

Identifies the text of one or more copyright notices for a software Package, File or Snippet, if any.

Description

A copyrightText consists of the text(s) of the copyright notice(s) found for a software Package, File or Snippet, if any.

If a copyrightText contains text, then it may contain any text related to one or more copyright notices (even if not complete) for that software Package, File or Snippet.

If a copyrightText has a “NONE” value, this indicates that the software Package, File or Snippet contains no copyright notice whatsoever.

If a copyrightText has a “NOASSERTION” value, this indicates that one of the following applies:

- the SPDX data creator has attempted to but cannot reach a reasonable objective determination;
- the SPDX data creator has made no attempt to determine this field; or
- the SPDX data creator has intentionally provided no information (no meaning should be implied by doing so).

If a copyrightText is present, but consists of solely an empty string or a string with no substantive content (e.g., a string that contains only whitespace), then this should be interpreted as equivalent to a “NOASSERTION” value as described above.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/copyrightText`

Name:	copyrightText
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Software/SoftwareArtifact

9.2.8 downloadLocation

Summary

Identifies the download Uniform Resource Identifier for the package at the time that the document was created.

Description

A downloadLocation identifies the download Uniform Resource Identifier for the package at the time that the document was created.

Where and how to download the exact package being referenced is critical for verification and tracking data.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/downloadLocation`

Name:	downloadLocation
Nature:	DataProperty
Range:	xsd:anyURI

Referenced

- /Software/Package

9.2.9 fileKind

Summary

Describes if a given file is a directory or non-directory kind of file.

Description

An SPDX file may represent a specific file or a directory of files.

In the future, this may be extended to other kinds (e.g. network based files).

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/fileKind`

Name:	fileKind
Nature:	ObjectProperty
Range:	FileKindType

Referenced

- /Software/File

9.2.10 homePage

Summary

A place for the SPDX document creator to record a website that serves as the package's home page.

Description

A homePage is a place for the SPDX document creator to record a website that serves as the package's home page.

This saves the recipient of the SPDX document who is looking for more info from having to search for and verify a match between the package and the associated project home page.

This link can also be used to reference further information about the package referenced by the SPDX document creator.

Metadata

```
https://spdx.org/rdf/3.0.1/terms/Software/homePage
```

Name:	homePage
Nature:	DataProperty
Range:	xsd:anyURI

Referenced

- /Software/Package

9.2.11 lineRange

Summary

Defines the line range in the original host file that the snippet information applies to.

Description

This field defines the line range in the original host file that the snippet information applies to.

If there is a disagreement between the byte range and line range, the byte range values will take precedence.

A range of lines is a convenient reference for those files where there is a known line delimiter. The choice was made to start the numbering of the lines at 1 to be consistent with the W3C pointer method vocabulary.

Metadata

```
https://spdx.org/rdf/3.0.1/terms/Software/lineRange
```

Name:	lineRange
Nature:	DataProperty
Range:	/Core/PositiveIntegerRange

Referenced

- /Software/Snippet

9.2.12 packageUrl

Summary

Provides a place for the SPDX data creator to record the package URL string (in accordance with the Package URL specification) for a software Package.

Description

A package URL (commonly pronounced and referred to as "purl") is an attempt to standardize package representations in order to reliably identify and locate software packages. A packageUrl is a URL string which represents a package in a mostly universal and uniform way across programming languages, package managers, packaging conventions, tools, APIs and databases.

A packageUrl is composed of seven components:

```
scheme:type/namespace/name@version?qualifiers#subpath
```

The definition for each component can be found in the corresponding Annex[56] of this specification. Known type definitions can be found in the Package URL type definitions[57].

[56] ../../../annexes/pkg-url-specification.md

[57] https://github.com/package-url/purl-spec/blob/b33dda1cf4515efa8eabbbe8e9b140950805f845/PURL-TYPES.rst

Components are designed such that they form a hierarchy from the most significant on the left to the least significant components on the right.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/packageUrl`

Name:	packageUrl
Nature:	DataProperty
Range:	xsd:anyURI

Referenced

- /Software/Package

9.2.13 packageVersion

Summary

Identify the version of a package.

Description

A packageVersion is useful for identification purposes and for indicating later changes of the package version.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/packageVersion`

Name:	packageVersion
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Software/Package

9.2.14 primaryPurpose

Summary

Provides information about the primary purpose of the software artifact.

Description

primaryPurpose provides information about the primary purpose of the software artifact.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/primaryPurpose`

Name:	primaryPurpose
Nature:	ObjectProperty
Range:	SoftwarePurpose

Referenced

- /Software/SoftwareArtifact

9.2.15 sbomType

Summary

Provides information about the type of an SBOM.

Description

This field is a reasonable estimation of the type of SBOM created from a creator perspective.

It is intended to be used to give guidance on the elements that may be contained within it.

Aligning with the guidance produced in Types of Software Bill of Material (SBOM) Documents[58].

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/sbomType`

Name:	sbomType
Nature:	ObjectProperty
Range:	SbomType

Referenced

- /Software/Sbom

9.2.16 snippetFromFile

Summary

Defines the original host file that the snippet information applies to.

Description

The field identifies the file which contains the snippet.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/snippetFromFile`

Name:	snippetFromFile
Nature:	ObjectProperty
Range:	File

Referenced

- /Software/Snippet

9.2.17 sourceInfo

Summary

Records any relevant background information or additional comments about the origin of the package.

Description

A sourceInfo records any relevant background information or additional comments about the origin of the package.

For example, this field might include comments indicating whether the package was pulled from a source code management system or has been repackaged.

The creator can provide additional information to describe any anomalies or discoveries in the determination of the origin of the package.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/sourceInfo`

Name:	sourceInfo
Nature:	DataProperty
Range:	xsd:string

[58] https://www.cisa.gov/sites/default/files/2023-04/sbom-types-document-508c.pdf

Referenced

- /Software/Package

9.3 Vocabularies

9.3.1 ContentIdentifierType

Summary

Specifies the type of a content identifier.

Description

ContentIdentifierType specifies the type of a content identifier.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/ContentIdentifierType`

Name:	ContentIdentifierType

Entries

gitoid Gitoid[59], stands for Git Object ID[60]. A gitoid of type blob is a unique hash of a binary artifact. A gitoid may represent either an Artifact Identifier[61] for the software artifact or an Input Manifest Identifier[62] for the software artifact's associated Artifact Input Manifest[63]; this ambiguity exists because the Artifact Input Manifest is itself an artifact, and the gitoid of that artifact is its valid identifier. Gitoids calculated on software artifacts (Snippet, File, or Package Elements) should be recorded in the SPDX 3.0 SoftwareArtifact's contentIdentifier property. Gitoids calculated on the Artifact Input Manifest (Input Manifest Identifier) should be recorded in the SPDX 3.0 Element's externalIdentifier property. See OmniBOR Specification[64], a minimalistic specification for describing software Artifact Dependency Graphs[65].

swhid SoftWare Hash IDentifier, a persistent intrinsic identifier for digital artifacts, such as files, trees (also known as directories or folders), commits, and other objects typically found in version control systems. The format of the identifiers is defined in the SWHID specification[66] (ISO/IEC DIS 18670). They typically look like `swh:1:cnt:94a9ed024d3859793618152ea559a168b`

9.3.2 FileKindType

Summary

Enumeration of the different kinds of SPDX file.

Description

An SPDX file may represent a file on disk or a directory of files.

In the future, this may be extended to other kinds (e.g. network based files).

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/FileKindType`

Name:	FileKindType

Entries

directory The file represents a directory and all content stored in that directory.

file The file represents a single file (default).

[59] https://www.iana.org/assignments/uri-schemes/prov/gitoid

[60] https://git-scm.com/book/en/v2/Git-Internals-Git-Objects

[61] https://github.com/omnibor/spec/blob/eb1ee5c961c16215eb8709b2975d193a2007a35d/spec/SPEC.md#artifact-identifier-types

[62] https://github.com/omnibor/spec/blob/eb1ee5c961c16215eb8709b2975d193a2007a35d/spec/SPEC.md#input-manifest-identifier

[63] https://github.com/omnibor/spec/blob/eb1ee5c961c16215eb8709b2975d193a2007a35d/spec/SPEC.md#artifact-input-manifest

[64] https://github.com/omnibor/spec/

[65] https://github.com/omnibor/spec/blob/eb1ee5c961c16215eb8709b2975d193a2007a35d/spec/SPEC.md#artifact-dependency-graph-adg

[66] https://www.swhid.org/specification/v1.1/4.Syntax

9.3.3 SbomType

Summary

Provides a set of values to be used to describe the common types of SBOMs that tools may create.

Description

The set of SBOM types with definitions as defined in Types of Software Bill of Material (SBOM) Documents[67], published on April 21, 2023.

An SBOM type describes the most likely type of an SBOM from the producer perspective, so that consumers can draw conclusions about the data inside an SBOM.

A single SBOM can have multiple SBOM document types associated with it.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/SbomType`

Name:	SbomType

Entries

analyzed SBOM generated through analysis of artifacts (e.g., executables, packages, containers, and virtual machine images) after its build. Such analysis generally requires a variety of heuristics. In some contexts, this may also be referred to as a “3rd party” SBOM.

build SBOM generated as part of the process of building the software to create a releasable artifact (e.g., executable or package) from data such as source files, dependencies, built components, build process ephemeral data, and other SBOMs.

deployed SBOM provides an inventory of software that is present on a system. This may be an assembly of other SBOMs that combines analysis of configuration options, and examination of execution behavior in a (potentially simulated) deployment environment.

design SBOM of intended, planned software project or product with included components (some of which may not yet exist) for a new software artifact.

runtime SBOM generated through instrumenting the system running the software, to capture only components present in the system, as well as external call-outs or dynamically loaded components. In some contexts, this may also be referred to as an “Instrumented” or “Dynamic” SBOM.

source SBOM created directly from the development environment, source files, and included dependencies used to build an product artifact.

9.3.4 SoftwarePurpose

Summary

Provides information about the primary purpose of an Element.

Description

This field provides information about the primary purpose of an Element.

Software Purpose is intrinsic to how the Element is being used rather than the content of the Element.

This field is a reasonable estimate of the most likely usage of the Element from the producer and consumer perspective from which both parties can draw conclusions about the context in which the Element exists.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Software/SoftwarePurpose`

Name:	SoftwarePurpose

[67] https://www.cisa.gov/sites/default/files/2023-04/sbom-types-document-508c.pdf

Entries

application The Element is a software application.

archive The Element is an archived collection of one or more files (.tar, .zip, etc.).

bom The Element is a bill of materials.

configuration The Element is configuration data.

container The Element is a container image which can be used by a container runtime application.

data The Element is data.

device The Element refers to a chipset, processor, or electronic board.

deviceDriver The Element represents software that controls hardware devices.

diskImage The Element refers to a disk image that can be written to a disk, booted in a VM, etc. A disk image typically contains most or all of the components necessary to boot, such as bootloaders, kernels, firmware, userspace, etc.

documentation The Element is documentation.

evidence The Element is the evidence that a specification or requirement has been fulfilled.

executable The Element is an Artifact that can be run on a computer.

file The Element is a single file which can be independently distributed (configuration file, statically linked binary, Kubernetes deployment, etc.).

filesystemImage The Element is a file system image that can be written to a disk (or virtual) partition.

firmware The Element provides low level control over a device's hardware.

framework The Element is a software framework.

install The Element is used to install software on disk.

library The Element is a software library.

manifest The Element is a software manifest.

model The Element is a machine learning or artificial intelligence model.

module The Element is a module of a piece of software.

operatingSystem The Element is an operating system.

other The Element doesn't fit into any of the other categories.

patch The Element contains a set of changes to update, fix, or improve another Element.

platform The Element represents a runtime environment.

requirement The Element provides a requirement needed as input for another Element.

source The Element is a single or a collection of source files.

specification The Element is a plan, guideline or strategy how to create, perform or analyze an application.

test The Element is a test used to verify functionality on an software element.

10 Security

Summary

The Security Profile captures security related information.

Description

The Security Profile captures security related information.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security`

Name:	Security

10.1 Classes

10.1.1 CvssV2VulnAssessmentRelationship

Summary

Provides a CVSS version 2.0 assessment for a vulnerability.

Description

A CvssV2VulnAssessmentRelationship relationship describes the determined score and vector of a vulnerability as defined in A Complete Guide to the Common Vulnerability Scoring System Version 2.0[68].

It is intended to communicate the results of using a CVSS calculator.

Constraints

- The relationship type must be set to `hasAssessmentFor`.

Example

```
{
  "type": "CvssV2VulnAssessmentRelationship",
  "spdxId": "urn:spdx.dev:cvssv2-cve-2020-28498",
  "relationshipType": "hasAssessmentFor",
  "security_score": "4.3",
  "security_vectorString": "(AV:N/AC:M/Au:N/C:P/I:N/A:N)",
  "from": "urn:spdx.dev:vuln-cve-2020-28498",
  "to": ["urn:product-acme-application-1.3"],
  "security_assessedElement": "urn:npm-elliptic-6.5.2",
  "externalRef": [
    {
      "type": "ExternalRef",
      "externalRefType": "securityAdvisory",
      "locator": "https://nvd.nist.gov/vuln/detail/CVE-2020-28498"
    },
    {
      "type": "ExternalRef",
      "externalRefType": "securityAdvisory",
      "locator": "https://snyk.io/vuln/SNYK-JS-ELLIPTIC-1064899"
    },
    {
      "type": "ExternalRef",
      "externalRefType": "securityFix",
      "locator": "https://github.com/indutny/elliptic/commit/441b742"
    }
  ],
  "suppliedBy": ["urn:spdx.dev:agent-my-security-vendor"],
  "publishedTime": "2023-05-06T10:06:13Z"
},
```

[68] https://www.first.org/cvss/v2/guide

```
{
  "type": "Relationship",
  "spdxId": "urn:spdx.dev:vulnAgentRel-1",
  "relationshipType": "publishedBy",
  "from": "urn:spdx.dev:cvssv2-cve-2020-28498",
  "to": ["urn:spdx.dev:agent-snyk"],
  "startTime": "2021-03-08T16:06:50Z"
}
```

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/CvssV2VulnAssessmentRelationship`

Name:	CvssV2VulnAssessmentRelationship
Instantiability:	Concrete
SubclassOf:	VulnAssessmentRelationship

Superclasses

- /Security/VulnAssessmentRelationship
- /Core/Relationship
- /Core/Element

Properties

Property	Type	minCount	maxCount
score	xsd:decimal	1	1
vectorString	xsd:string	1	1

All properties (informative)

Property	Type	minCount	maxCount
assessedElement	/Software/SoftwareArtifact	0	1
comment	xsd:string	0	1
completeness	RelationshipCompleteness	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
endTime	DateTime	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
from	Element	1	1
modifiedTime	/Core/DateTime	0	1
name	xsd:string	0	1
publishedTime	/Core/DateTime	0	1
relationshipType	RelationshipType	1	1
score	xsd:decimal	1	1
spdxId	xsd:anyURI	1	1
startTime	DateTime	0	1
summary	xsd:string	0	1
suppliedBy	/Core/Agent	0	1
to	Element	1	*
vectorString	xsd:string	1	1
verifiedUsing	IntegrityMethod	0	*
withdrawnTime	/Core/DateTime	0	1

10.1.2 CvssV3VulnAssessmentRelationship

Summary

Provides a CVSS version 3 assessment for a vulnerability.

Description

A CvssV3VulnAssessmentRelationship relationship describes the determined score, severity, and vector of a vulnerability as defined in Common Vulnerability Scoring System v3.0: Specification Document[69] or Common Vulnerability Scoring System v3.1: Specification Document[70].

It is intended to communicate the results of using a CVSS calculator.

Constraints

- The relationship type must be set to `hasAssessmentFor`.

Example

```
{
  "type": "CvssV3VulnAssessmentRelationship",
  "spdxId": "urn:spdx.dev:cvssv3-cve-2020-28498",
  "relationshipType": "hasAssessmentFor",
  "security_score": "6.8",
  "security_severity": "medium",
  "security_vectorString": "CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:C/C:H/I:N/A:N",
  "from": "urn:spdx.dev:vuln-cve-2020-28498",
  "to": ["urn:product-acme-application-1.3"],
  "security_assessedElement": "urn:npm-elliptic-6.5.2",
  "externalRef": [
    {
      "type": "ExternalRef",
      "externalRefType": "securityAdvisory",
      "locator": "https://nvd.nist.gov/vuln/detail/CVE-2020-28498"
    },
    {
      "type": "ExternalRef",
      "externalRefType": "securityAdvisory",
      "locator": "https://snyk.io/vuln/SNYK-JS-ELLIPTIC-1064899"
    },
    {
      "type": "ExternalRef",
      "externalRefType": "securityFix",
      "locator": "https://github.com/indutny/elliptic/commit/441b742"
    }
  ],
  "suppliedBy": ["urn:spdx.dev:agent-my-security-vendor"],
  "publishedTime": "2023-05-06T10:06:13Z"
},
{
  "type": "Relationship",
  "spdxId": "urn:spdx.dev:vulnAgentRel-1",
  "relationshipType": "publishedBy",
  "from": "urn:spdx.dev:cvssv3-cve-2020-28498",
  "to": ["urn:spdx.dev:agent-snyk"],
  "startTime": "2021-03-08T16:06:50Z"
}
```

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/CvssV3VulnAssessmentRelationship`

Name:	CvssV3VulnAssessmentRelationship
Instantiability:	Concrete
SubclassOf:	VulnAssessmentRelationship

[69] https://www.first.org/cvss/v3.0/specification-document

[70] https://www.first.org/cvss/v3.1/specification-document

Superclasses

- /Security/VulnAssessmentRelationship
- /Core/Relationship
- /Core/Element

Properties

Property	Type	minCount	maxCount
score	xsd:decimal	1	1
severity	CvssSeverityType	1	1
vectorString	xsd:string	1	1

All properties (informative)

Property	Type	minCount	maxCount
assessedElement	/Software/SoftwareArtifact	0	1
comment	xsd:string	0	1
completeness	RelationshipCompleteness	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
endTime	DateTime	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
from	Element	1	1
modifiedTime	/Core/DateTime	0	1
name	xsd:string	0	1
publishedTime	/Core/DateTime	0	1
relationshipType	RelationshipType	1	1
score	xsd:decimal	1	1
severity	CvssSeverityType	1	1
spdxId	xsd:anyURI	1	1
startTime	DateTime	0	1
summary	xsd:string	0	1
suppliedBy	/Core/Agent	0	1
to	Element	1	*
vectorString	xsd:string	1	1
verifiedUsing	IntegrityMethod	0	*
withdrawnTime	/Core/DateTime	0	1

10.1.3 CvssV4VulnAssessmentRelationship

Summary

Provides a CVSS version 4 assessment for a vulnerability.

Description

A CvssV4VulnAssessmentRelationship relationship describes the determined score, severity, and vector of a vulnerability as defined in Common Vulnerability Scoring System version 4.0: Specification Document[71].

It is intended to communicate the results of using a CVSS calculator.

Constraints

- The relationship type must be set to hasAssessmentFor.

Example

```
{
  "type": "CvssV4VulnAssessmentRelationship",
```

[71] https://www.first.org/cvss/v4.0/specification-document

```
    "spdxId": "urn:spdx.dev:cvssv4-cve-2021-44228",
    "relationshipType": "hasAssessmentFor",
    "security_severity": "medium",
    "security_score": "10.0",
    "security_vectorString": "CVSS:4.0/AV:N/AC:L/AT:N/AR:N/UI:N/VCH/VI:H/VA:H/SC:H/SI:H/SA:H/E:A",
    "from": "urn:spdx.dev:vuln-cve-2021-44228",
    "to": ["urn:product-acme-application-1.3"],
    "security_assessedElement": "urn:apache-log4j-2.14.1",
    "externalRef": [
      {
        "@type": "ExternalRef",
        "externalRefType": "securityAdvisory",
        "locator": "https://nvd.nist.gov/vuln/detail/CVE-2021-44228"
      },
      {
        "@type": "ExternalRef",
        "externalRefType": "securityAdvisory",
        "locator": "https://logging.apache.org/log4j/2.x/security.html"
      },
      {
        "@type": "ExternalRef",
        "externalRefType": "securityOther",
        "locator": "https://www.first.org/cvss/v4.0/examples#Apache-log4j-Vulnerability-CVE-2021-44228"
      },
    ],
    "suppliedBy": ["urn:spdx.dev:agent-my-security-vendor"],
    "publishedTime": "2023-10-05T23:09:13Z"
  },
  {
    "type": "Relationship",
    "spdxId": "urn:spdx.dev:vulnAgentRel-1",
    "relationshipType": "publishedBy",
    "from": "urn:spdx.dev:cvssv4-cve-2021-44228",
    "to": ["urn:spdx.dev:agent-apache.org"],
    "startTime": "2021-12-11T18:39:00Z"
  }
```

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/CvssV4VulnAssessmentRelationship`

Name:	CvssV4VulnAssessmentRelationship
Instantiability:	Concrete
SubclassOf:	VulnAssessmentRelationship

Superclasses

- /Security/VulnAssessmentRelationship
- /Core/Relationship
- /Core/Element

Properties

Property	Type	minCount	maxCount
score	xsd:decimal	1	1
severity	CvssSeverityType	1	1
vectorString	xsd:string	1	1

All properties (informative)

Property	Type	minCount	maxCount
assessedElement	/Software/SoftwareArtifact	0	1
comment	xsd:string	0	1
completeness	RelationshipCompleteness	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
endTime	DateTime	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
from	Element	1	1
modifiedTime	/Core/DateTime	0	1
name	xsd:string	0	1
publishedTime	/Core/DateTime	0	1
relationshipType	RelationshipType	1	1
score	xsd:decimal	1	1
severity	CvssSeverityType	1	1
spdxId	xsd:anyURI	1	1
startTime	DateTime	0	1
summary	xsd:string	0	1
suppliedBy	/Core/Agent	0	1
to	Element	1	*
vectorString	xsd:string	1	1
verifiedUsing	IntegrityMethod	0	*
withdrawnTime	/Core/DateTime	0	1

10.1.4 EpssVulnAssessmentRelationship

Summary

Provides an EPSS assessment for a vulnerability.

Description

An EpssVulnAssessmentRelationship relationship describes the likelihood or probability that a vulnerability will be exploited in the wild, and the percentile ranking of probability relative to all other vulnerabilities' EPSS scores, using the Exploit Prediction Scoring System (EPSS) as defined at The EPSS Model[72].

Constraints

- The relationship type must be set to hasAssessmentFor.
- The probability must be between 0 and 1.
- The percentile must be between 0 and 1.

Example

```
{
  "type": "EpssVulnAssessmentRelationship",
  "spdxId": "urn:spdx.dev:epss-CVE-2020-28498",
  "relationshipType": "hasAssessmentFor",
  "security_probability": "0.00105",
  "security_percentile": "0.42356",
  "from": "urn:spdx.dev:vuln-cve-2020-28498",
  "to": ["urn:product-acme-application-1.3"],
  "suppliedBy": ["urn:spdx.dev:agent-jane-doe"],
  "publishedTime": "2023-10-05T00:00:30Z"
}
```

[72] https://www.first.org/epss/model

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/EpssVulnAssessmentRelationship`

Name:	EpssVulnAssessmentRelationship
Instantiability:	Concrete
SubclassOf:	VulnAssessmentRelationship

Superclasses

- /Security/VulnAssessmentRelationship
- /Core/Relationship
- /Core/Element

Properties

Property	Type	minCount	maxCount
percentile	xsd:decimal	1	1
probability	xsd:decimal	1	1

External properties cardinality updates

Property	minCount	maxCount
/Security/VulnAssessmentRelationship/publishedTime	1	

All properties (informative)

Property	Type	minCount	maxCount
assessedElement	/Software/SoftwareArtifact	0	1
comment	xsd:string	0	1
completeness	RelationshipCompleteness	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
endTime	DateTime	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
from	Element	1	1
modifiedTime	/Core/DateTime	0	1
name	xsd:string	0	1
percentile	xsd:decimal	1	1
probability	xsd:decimal	1	1
publishedTime	/Core/DateTime	1	1
relationshipType	RelationshipType	1	1
spdxId	xsd:anyURI	1	1
startTime	DateTime	0	1
summary	xsd:string	0	1
suppliedBy	/Core/Agent	0	1
to	Element	1	*
verifiedUsing	IntegrityMethod	0	*
withdrawnTime	/Core/DateTime	0	1

10.1.5 ExploitCatalogVulnAssessmentRelationship

Summary

Provides an exploit assessment of a vulnerability.

Description

An ExploitCatalogVulnAssessmentRelationship describes if a vulnerability is listed in any exploit catalog such as the CISA Known Exploited Vulnerabilities (KEV) Catalog[73].

[73] https://www.cisa.gov/known-exploited-vulnerabilities-catalog

Constraints

- The relationship type must be set to hasAssessmentFor.

Example

```
{
  "type": "ExploitCatalogVulnAssessmentRelationship",
  "spdxId": "urn:spdx.dev:exploit-catalog-1",
  "relationshipType": "hasAssessmentFor",
  "security_catalogType": "kev",
  "locator": "https://www.cisa.gov/known-exploited-vulnerabilities-catalog",
  "security_exploited": "true",
  "from": "urn:spdx.dev:vuln-cve-2023-2136",
  "to": ["urn:product-google-chrome-112.0.5615.136"],
  "suppliedBy": ["urn:spdx.dev:agent-jane-doe"],
  "publishedTime": "2021-03-09T11:04:53Z"
}
```

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/ExploitCatalogVulnAssessmentRelationship`

Name:	ExploitCatalogVulnAssessmentRelationship
Instantiability:	Concrete
SubclassOf:	VulnAssessmentRelationship

Superclasses

- /Security/VulnAssessmentRelationship
- /Core/Relationship
- /Core/Element

Properties

Property	Type	minCount	maxCount
catalogType	ExploitCatalogType	1	1
exploited	xsd:boolean	1	1
locator	xsd:anyURI	1	1

All properties (informative)

Property	Type	minCount	maxCount
assessedElement	/Software/SoftwareArtifact	0	1
catalogType	ExploitCatalogType	1	1
comment	xsd:string	0	1
completeness	RelationshipCompleteness	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
endTime	DateTime	0	1
exploited	xsd:boolean	1	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
from	Element	1	1
locator	xsd:anyURI	1	1
modifiedTime	/Core/DateTime	0	1
name	xsd:string	0	1
publishedTime	/Core/DateTime	0	1
relationshipType	RelationshipType	1	1
spdxId	xsd:anyURI	1	1
startTime	DateTime	0	1
summary	xsd:string	0	1
suppliedBy	/Core/Agent	0	1
to	Element	1	*
verifiedUsing	IntegrityMethod	0	*
withdrawnTime	/Core/DateTime	0	1

10.1.6 SsvcVulnAssessmentRelationship

Summary

Provides an SSVC assessment for a vulnerability.

Description

An SsvcVulnAssessmentRelationship describes the decision made using the Stakeholder-Specific Vulnerability Categorization (SSVC) decision tree as defined by CISA Stakeholder-Specific Vulnerability Categorization Guide[74].

It is intended to communicate the results of using the CISA SSVC Calculator.

Constraints

- The relationship type must be set to hasAssessmentFor.

Example

```
{
  "@type": "SsvcVulnAssessmentRelationship",
  "@id": "urn:spdx.dev:ssvc-1",
  "relationshipType": "hasAssessmentFor",
  "security_decisionType": "act",
  "from": "urn:spdx.dev:vuln-cve-2020-28498",
  "to": ["urn:product-acme-application-1.3"],
  "security_assessedElement": "urn:npm-elliptic-6.5.2",
  "suppliedBy": ["urn:spdx.dev:agent-jane-doe"],
  "publishedTime": "2021-03-09T11:04:53Z"
}
```

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/SsvcVulnAssessmentRelationship`

[74] https://www.cisa.gov/stakeholder-specific-vulnerability-categorization-ssvc

Name:	SsvcVulnAssessmentRelationship
Instantiability:	Concrete
SubclassOf:	VulnAssessmentRelationship

Superclasses

- /Security/VulnAssessmentRelationship
- /Core/Relationship
- /Core/Element

Properties

Property	Type	minCount	maxCount
decisionType	SsvcDecisionType	1	1

All properties (informative)

Property	Type	minCount	maxCount
assessedElement	/Software/SoftwareArtifact	0	1
comment	xsd:string	0	1
completeness	RelationshipCompleteness	0	1
creationInfo	CreationInfo	1	1
decisionType	SsvcDecisionType	1	1
description	xsd:string	0	1
endTime	DateTime	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
from	Element	1	1
modifiedTime	/Core/DateTime	0	1
name	xsd:string	0	1
publishedTime	/Core/DateTime	0	1
relationshipType	RelationshipType	1	1
spdxId	xsd:anyURI	1	1
startTime	DateTime	0	1
summary	xsd:string	0	1
suppliedBy	/Core/Agent	0	1
to	Element	1	*
verifiedUsing	IntegrityMethod	0	*
withdrawnTime	/Core/DateTime	0	1

10.1.7 VexAffectedVulnAssessmentRelationship

Summary

Connects a vulnerability and an element designating the element as a product affected by the vulnerability.

Description

VexAffectedVulnAssessmentRelationship connects a vulnerability and a number of elements. The relationship marks these elements as products affected by the vulnerability. This relationship corresponds to the VEX affected status.

Constraints

When linking elements using a VexAffectedVulnAssessmentRelationship, the following requirements must be observed:

- Elements linked with a VulnVexAffectedAssessmentRelationship are constrained to the affects relationship type.

Example

```
{
  "type": "VexAffectedVulnAssessmentRelationship",
  "spdxId": "urn:spdx.dev:vex-affected-1",
  "relationshipType": "affects",
  "from": "urn:spdx.dev:vuln-cve-2020-28498",
  "to": ["urn:product-acme-application-1.3"],
  "security_assessedElement": "urn:npm-elliptic-6.5.2",
  "security_actionStatement": "Upgrade to version 1.4 of ACME application.",
  "suppliedBy": ["urn:spdx.dev:agent-jane-doe"],
  "publishedTime": "2021-03-09T11:04:53Z"
}
```

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/VexAffectedVulnAssessmentRelationship`

Name:	VexAffectedVulnAssessmentRelationship
Instantiability:	Concrete
SubclassOf:	VexVulnAssessmentRelationship

Superclasses

- /Security/VexVulnAssessmentRelationship
- /Security/VulnAssessmentRelationship
- /Core/Relationship
- /Core/Element

Properties

Property	Type	minCount	maxCount
actionStatement	xsd:string	1	1
actionStatementTime	/Core/DateTime	0	1

All properties (informative)

Property	Type	minCount	maxCount
actionStatement	xsd:string	1	1
actionStatementTime	/Core/DateTime	0	1
assessedElement	/Software/SoftwareArtifact	0	1
comment	xsd:string	0	1
completeness	RelationshipCompleteness	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
endTime	DateTime	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
from	Element	1	1
modifiedTime	/Core/DateTime	0	1
name	xsd:string	0	1
publishedTime	/Core/DateTime	0	1
relationshipType	RelationshipType	1	1
spdxId	xsd:anyURI	1	1
startTime	DateTime	0	1
statusNotes	xsd:string	0	1
summary	xsd:string	0	1
suppliedBy	/Core/Agent	0	1
to	Element	1	*
verifiedUsing	IntegrityMethod	0	*
vexVersion	xsd:string	0	1
withdrawnTime	/Core/DateTime	0	1

10.1.8 VexFixedVulnAssessmentRelationship

Summary

Links a vulnerability and elements representing products (in the VEX sense) where a fix has been applied and are no longer affected.

Description

VexFixedVulnAssessmentRelationship links a vulnerability to a number of elements representing VEX products where a vulnerability has been fixed and are no longer affected. It represents the VEX fixed status.

Constraints

When linking elements using a VexFixedVulnAssessmentRelationship, the following requirements must be observed:

- Elements linked with a VulnVexFixedAssessmentRelationship are constrained to using the fixedIn relationship type.
- The from: end of the relationship must be a /Security/Vulnerability classed element.

Example

```
{
  "type": "VexFixedVulnAssessmentRelationship",
  "spdxId": "urn:spdx.dev:vex-fixed-in-1",
  "relationshipType": "fixedIn",
  "from": "urn:spdx.dev:vuln-cve-2020-28498",
  "to": ["urn:product-acme-application-1.3"],
  "security_assessedElement": "urn:npm-elliptic-6.5.4",
  "suppliedBy": ["urn:spdx.dev:agent-jane-doe"],
  "publishedTime": "2021-03-09T11:04:53Z"
}
```

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/VexFixedVulnAssessmentRelationship`

Name:	VexFixedVulnAssessmentRelationship
Instantiability:	Concrete
SubclassOf:	VexVulnAssessmentRelationship

Superclasses

- /Security/VexVulnAssessmentRelationship
- /Security/VulnAssessmentRelationship
- /Core/Relationship
- /Core/Element

All properties (informative)

Property	Type	minCount	maxCount
assessedElement	/Software/SoftwareArtifact	0	1
comment	xsd:string	0	1
completeness	RelationshipCompleteness	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
endTime	DateTime	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
from	Element	1	1
modifiedTime	/Core/DateTime	0	1
name	xsd:string	0	1
publishedTime	/Core/DateTime	0	1
relationshipType	RelationshipType	1	1
spdxId	xsd:anyURI	1	1
startTime	DateTime	0	1
statusNotes	xsd:string	0	1
summary	xsd:string	0	1
suppliedBy	/Core/Agent	0	1
to	Element	1	*
verifiedUsing	IntegrityMethod	0	*
vexVersion	xsd:string	0	1
withdrawnTime	/Core/DateTime	0	1

10.1.9 VexNotAffectedVulnAssessmentRelationship

Summary

Links a vulnerability and one or more elements designating the latter as products not affected by the vulnerability.

Description

VexNotAffectedVulnAssessmentRelationship connects a vulnerability and a number of elements designating them as products not affected by the vulnerability. This relationship corresponds to the VEX not_affected status.

Constraints

When linking elements using a VexNotVulnAffectedAssessmentRelationship, the following requirements must be observed:

- Relating elements with a VexNotAffectedVulnAssessmentRelationship is restricted to the doesNotAffect relationship type.
- The from: end of the relationship must be a /Security/Vulnerability classed element.
- Both impactStatement and justificationType properties have a cardinality of 0..1 making them optional. Nevertheless, to produce a valid VEX not_affected statement, one of them MUST be defined. This is specified in the Minimum Elements for VEX.

Example

```
{
  "type": "VexNotAffectedVulnAssessmentRelationship",
  "spdxId": "urn:spdx.dev:vex-not-affected-1",
  "relationshipType": "doesNotAffect",
  "from": "urn:spdx.dev:vuln-cve-2020-28498",
  "to": ["urn:product-acme-application-1.3"],
  "security_assessedElement": "urn:npm-elliptic-6.5.2",
  "security_justificationType": "componentNotPresent",
  "security_impactStatement": "Not using this vulnerable part of this library.",
  "suppliedBy": ["urn:spdx.dev:agent-jane-doe"],
  "publishedTime": "2021-03-09T11:04:53Z"
}
```

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/VexNotAffectedVulnAssessmentRelationship`

Name:	VexNotAffectedVulnAssessmentRelationship
Instantiability:	Concrete
SubclassOf:	VexVulnAssessmentRelationship

Superclasses

- /Security/VexVulnAssessmentRelationship
- /Security/VulnAssessmentRelationship
- /Core/Relationship
- /Core/Element

Properties

Property	Type	minCount	maxCount
impactStatement	xsd:string	0	1
impactStatementTime	/Core/DateTime	0	1
justificationType	VexJustificationType	0	1

All properties (informative)

Property	Type	minCount	maxCount
assessedElement	/Software/SoftwareArtifact	0	1
comment	xsd:string	0	1
completeness	RelationshipCompleteness	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
endTime	DateTime	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
from	Element	1	1
impactStatement	xsd:string	0	1
impactStatementTime	/Core/DateTime	0	1
justificationType	VexJustificationType	0	1
modifiedTime	/Core/DateTime	0	1
name	xsd:string	0	1
publishedTime	/Core/DateTime	0	1
relationshipType	RelationshipType	1	1
spdxId	xsd:anyURI	1	1
startTime	DateTime	0	1
statusNotes	xsd:string	0	1
summary	xsd:string	0	1
suppliedBy	/Core/Agent	0	1
to	Element	1	*
verifiedUsing	IntegrityMethod	0	*
vexVersion	xsd:string	0	1
withdrawnTime	/Core/DateTime	0	1

10.1.10 VexUnderInvestigationVulnAssessmentRelationship

Summary

Designates elements as products where the impact of a vulnerability is being investigated.

Description

VexUnderInvestigationVulnAssessmentRelationship links a vulnerability to a number of products stating the vulnerability's impact on them is being investigated. It represents the VEX under_investigation status.

Constraints

When linking elements using a VexUnderInvestigationVulnAssessmentRelationship the following requirements must be observed:

- Elements linked with a VexUnderInvestigationVulnAssessmentRelationship are constrained to using the underInvestigationFor relationship type.
- The from: end of the relationship must be a /Security/Vulnerability classed element.

Example

```
{
  "type": "VexUnderInvestigationVulnAssessmentRelationship",
  "spdxId": "urn:spdx.dev:vex-underInvestigation-1",
  "relationshipType": "underInvestigationFor",
  "from": "urn:spdx.dev:vuln-cve-2020-28498",
  "to": ["urn:product-acme-application-1.3"],
  "security_assessedElement": "urn:npm-elliptic-6.5.2",
  "suppliedBy": ["urn:spdx.dev:agent-jane-doe"],
  "publishedTime": "2021-03-09T11:04:53Z"
}
```

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/VexUnderInvestigationVulnAssessmentRelationship`

Name:	VexUnderInvestigationVulnAssessmentRelationship
Instantiability:	Concrete
SubclassOf:	VexVulnAssessmentRelationship

Superclasses

- /Security/VexVulnAssessmentRelationship
- /Security/VulnAssessmentRelationship
- /Core/Relationship
- /Core/Element

All properties (informative)

Property	Type	minCount	maxCount
assessedElement	/Software/SoftwareArtifact	0	1
comment	xsd:string	0	1
completeness	RelationshipCompleteness	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
endTime	DateTime	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
from	Element	1	1
modifiedTime	/Core/DateTime	0	1
name	xsd:string	0	1
publishedTime	/Core/DateTime	0	1
relationshipType	RelationshipType	1	1
spdxId	xsd:anyURI	1	1
startTime	DateTime	0	1
statusNotes	xsd:string	0	1
summary	xsd:string	0	1
suppliedBy	/Core/Agent	0	1
to	Element	1	*
verifiedUsing	IntegrityMethod	0	*
vexVersion	xsd:string	0	1
withdrawnTime	/Core/DateTime	0	1

10.1.11 VexVulnAssessmentRelationship

Summary

Abstract ancestor class for all VEX relationships

Description

VexVulnAssessmentRelationship is an abstract subclass that defined the common properties shared by all the SPDX-VEX status relationships.

Constraints

When linking elements using a VexVulnAssessmentRelationship, the following requirements must be observed:

- The from: end must be a /Security/Vulnerability classed element
- The to: end must point to elements representing the VEX *products*.

To specify a different element where the vulnerability was detected, the VEX relationship can optionally specify *subcomponents* using the assessedElement property.

VEX inherits information from the document level down to its statements. When a statement is missing information it can be completed by reading the equivalent field from the containing document. For example, if a VEX relationship is missing data in its createdBy property, tools must consider the entity listed in the CreationInfo section of the document as the VEX author. In the same way, when a VEX relationship does not have a created property, the document's date must be considered as authoritative.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/VexVulnAssessmentRelationship`

Name:	VexVulnAssessmentRelationship
Instantiability:	Abstract
SubclassOf:	VulnAssessmentRelationship

Superclasses

- /Security/VulnAssessmentRelationship
- /Core/Relationship
- /Core/Element

Properties

Property	Type	minCount	maxCount
statusNotes	xsd:string	0	1
vexVersion	xsd:string	0	1

All properties (informative)

Property	Type	minCount	maxCount
assessedElement	/Software/SoftwareArtifact	0	1
comment	xsd:string	0	1
completeness	RelationshipCompleteness	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
endTime	DateTime	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
from	Element	1	1
modifiedTime	/Core/DateTime	0	1
name	xsd:string	0	1
publishedTime	/Core/DateTime	0	1
relationshipType	RelationshipType	1	1
spdxId	xsd:anyURI	1	1
startTime	DateTime	0	1
statusNotes	xsd:string	0	1
summary	xsd:string	0	1
suppliedBy	/Core/Agent	0	1
to	Element	1	*
verifiedUsing	IntegrityMethod	0	*
vexVersion	xsd:string	0	1
withdrawnTime	/Core/DateTime	0	1

10.1.12 VulnAssessmentRelationship

Summary

Abstract ancestor class for all vulnerability assessments

Description

VulnAssessmentRelationship is the ancestor class common to all vulnerability assessment relationships. It factors out the common properties shared by them.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/VulnAssessmentRelationship`

Name:	VulnAssessmentRelationship
Instantiability:	Abstract
SubclassOf:	/Core/Relationship

Superclasses

- /Core/Relationship
- /Core/Element

Properties

Property	Type	minCount	maxCount
/Core/suppliedBy	/Core/Agent	0	1
assessedElement	/Software/SoftwareArtifact	0	1
modifiedTime	/Core/DateTime	0	1
publishedTime	/Core/DateTime	0	1
withdrawnTime	/Core/DateTime	0	1

All properties (informative)

Property	Type	minCount	maxCount
assessedElement	/Software/SoftwareArtifact	0	1
comment	xsd:string	0	1
completeness	RelationshipCompleteness	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
endTime	DateTime	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
from	Element	1	1
modifiedTime	/Core/DateTime	0	1
name	xsd:string	0	1
publishedTime	/Core/DateTime	0	1
relationshipType	RelationshipType	1	1
spdxId	xsd:anyURI	1	1
startTime	DateTime	0	1
summary	xsd:string	0	1
suppliedBy	/Core/Agent	0	1
to	Element	1	*
verifiedUsing	IntegrityMethod	0	*
withdrawnTime	/Core/DateTime	0	1

10.1.13 Vulnerability

Summary

Specifies a vulnerability and its associated information.

Description

Specifies a vulnerability and its associated information.

Example

```
{
  "type": "Vulnerability",
  "spdxId": "urn:spdx.dev:vuln-1",
  "summary": "Use of a Broken or Risky Cryptographic Algorithm",
  "description": "The package `elliptic` before version 6.5.4 are vulnerable to ..."
  "modifiedTime": "2021-03-08T16:06:43Z",
  "publishedTime": "2021-03-08T16:02:50Z",
  "externalIdentifier": [
    {
      "type": "ExternalIdentifier",
      "externalIdentifierType": "cve",
      "identifier": "CVE-2020-2849",
      "identifierLocator": [
        "https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-28498",
        "https://www.cve.org/CVERecord?id=CVE-2020-28498"
      ],
      "issuingAuthority": "urn:spdx.dev:agent-cve.org"
    },
    {
      "type": "ExternalIdentifier",
      "externalIdentifierType": "securityOther",
      "identifier": "GHSA-r9p9-mrjm-926w",
      "identifierLocator": "https://github.com/advisories/GHSA-r9p9-mrjm-926w"
    },
    {
      "type": "ExternalIdentifier",
      "externalIdentifierType": "securityOther",
      "identifier": "SNYK-JS-ELLIPTIC-1064899",
      "identifierLocator": "https://security.snyk.io/vuln/SNYK-JS-ELLIPTIC-1064899"
    }
  ],
  "externalRef": [
    {
      "type": "ExternalRef",
      "externalRefType": "securityAdvisory",
      "locator": "https://nvd.nist.gov/vuln/detail/CVE-2020-28498"
    },
    {
      "type": "ExternalRef",
      "externalRefType": "securityAdvisory",
      "locator": "https://ubuntu.com/security/CVE-2020-28498"
    },
    {
      "type": "ExternalRef",
      "externalRefType": "securityOther",
      "locator": "https://github.com/indutny/elliptic/pull/244/commits"
    },
    {
      "type": "ExternalRef",
      "externalRefType": "securityOther",
      "locator": "https://github.com/christianlundkvist/blog/2020_05_26_secp256k1_twist_attacks.md"
    }
  ]
},
{
  "type": "Relationship",
  "spdxId": "urn:spdx.dev:vulnRelationship-1",
  "relationshipType": "hasAssociatedVulnerability",
  "from": "urn:npm-elliptic-6.5.2",
  "to": ["urn:spdx.dev:vuln-1"],
  "startTime": "2021-03-08T16:06:50Z"
},
{
  "type": "Relationship",
  "spdxId": "urn:spdx.dev:vulnAgentRel-1",
  "relationshipType": "publishedBy",
  "from": "urn:spdx.dev:vuln-1",
  "to": ["urn:spdx.dev:agent-snyk"],
  "startTime": "2021-03-08T16:06:50Z"
}
```

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/Vulnerability`

Name:	Vulnerability
Instantiability:	Concrete
SubclassOf:	/Core/Artifact

Superclasses

- /Core/Artifact
- /Core/Element

Properties

Property	Type	minCount	maxCount
modifiedTime	/Core/DateTime	0	1
publishedTime	/Core/DateTime	0	1
withdrawnTime	/Core/DateTime	0	1

All properties (informative)

Property	Type	minCount	maxCount
builtTime	DateTime	0	1
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
modifiedTime	/Core/DateTime	0	1
name	xsd:string	0	1
originatedBy	Agent	0	*
publishedTime	/Core/DateTime	0	1
releaseTime	DateTime	0	1
spdxId	xsd:anyURI	1	1
standardName	xsd:string	0	*
summary	xsd:string	0	1
suppliedBy	Agent	0	1
supportLevel	SupportType	0	*
validUntilTime	DateTime	0	1
verifiedUsing	IntegrityMethod	0	*
withdrawnTime	/Core/DateTime	0	1

10.2 Properties

10.2.1 actionStatement

Summary

Provides advise on how to mitigate or remediate a vulnerability when a VEX product is affected by it.

Description

When an element is referenced with a VexAffectedVulnAssessmentRelationship, the relationship MUST include one actionStatement that SHOULD describe actions to remediate or mitigate the vulnerability.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/actionStatement`

Name:	actionStatement
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Security/VexAffectedVulnAssessmentRelationship

10.2.2 actionStatementTime

Summary

Records the time when a recommended action was communicated in a VEX statement to mitigate a vulnerability.

Description

When a VEX statement communicates an affected status, the author MUST include an action statement with a recommended action to help mitigate the vulnerability's impact. The actionStatementTime property records the time when the action statement was first communicated.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/actionStatementTime`

Name:	actionStatementTime
Nature:	DataProperty
Range:	/Core/DateTime

Referenced

- /Security/VexAffectedVulnAssessmentRelationship

10.2.3 assessedElement

Summary

Specifies an Element contained in a piece of software where a vulnerability was found.

Description

Specifies subpackages, files or snippets referenced by a security assessment to specify the precise location where a vulnerability was found.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/assessedElement`

Name:	assessedElement
Nature:	ObjectProperty
Range:	/Software/SoftwareArtifact

Referenced

- /Security/VulnAssessmentRelationship

10.2.4 catalogType

Summary

Specifies the exploit catalog type.

Description

A catalogType is a mandatory value and must select one of the existing entries in the `ExploitCatalogType`[75] vocabulary.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/catalogType`

Name:	catalogType
Nature:	ObjectProperty
Range:	ExploitCatalogType

[75] ../Vocabularies/ExploitCatalogType.md

Referenced

- /Security/ExploitCatalogVulnAssessmentRelationship

10.2.5 decisionType

Summary

Provide the enumeration of possible decisions in the [Stakeholder-Specific Vulnerability Categorization (SSVC) decision tree](https://www.cisa.gov/stakeholder-specific-vulnerability-categorization-ssvc).

Description

A decisionType is a mandatory value and must select one of the four entries in the `SsvcDecisionType`[76] vocabulary.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/decisionType`

Name:	decisionType
Nature:	ObjectProperty
Range:	SsvcDecisionType

Referenced

- /Security/SsvcVulnAssessmentRelationship

10.2.6 exploited

Summary

Describe that a CVE is known to have an exploit because it's been listed in an exploit catalog.

Description

This field is set when a CVE is listed in an exploit catalog.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/exploited`

Name:	exploited
Nature:	DataProperty
Range:	xsd:boolean

Referenced

- /Security/ExploitCatalogVulnAssessmentRelationship

10.2.7 impactStatement

Summary

Explains why a VEX product is not affected by a vulnerability. It is an alternative in VexNotAffectedVulnAssessmentRelationship to the machine-readable justification label.

Description

When a VEX product element is related with a VexNotAffectedVulnAssessmentRelationship and a machine readable justification label is not provided, then an impactStatement that further explains how or why the product(s) are not affected by the vulnerability must be provided.

[76] ../Vocabularies/SsvcDecisionType.md

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/impactStatement`

Name:	impactStatement
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Security/VexNotAffectedVulnAssessmentRelationship

10.2.8 impactStatementTime

Summary

Timestamp of impact statement.

Description

Specifies the time when the impact statement was recorded.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/impactStatementTime`

Name:	impactStatementTime
Nature:	DataProperty
Range:	/Core/DateTime

Referenced

- /Security/VexNotAffectedVulnAssessmentRelationship

10.2.9 justificationType

Summary

Impact justification label to be used when linking a vulnerability to an element representing a VEX product with a VexNotAffectedVulnAssessmentRelationship relationship.

Description

When stating that an element is not affected by a vulnerability, the VexNotAffectedVulnAssessmentRelationship must include a justification from the machine-readable labels catalog informing the reason the element is not impacted.

impactStatement which is a string with English prose can be used instead or as complementary to the justification label, but one of both MUST be defined.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/justificationType`

Name:	justificationType
Nature:	ObjectProperty
Range:	VexJustificationType

Referenced

- /Security/VexNotAffectedVulnAssessmentRelationship

10.2.10 locator

Summary

Provides the location of an exploit catalog.

Description

A locator provides the location of an exploit catalog.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/locator`

Name:	locator
Nature:	DataProperty
Range:	xsd:anyURI

Referenced

- /Security/ExploitCatalogVulnAssessmentRelationship

10.2.11 modifiedTime

Summary

Specifies a time when a vulnerability assessment was modified

Description

Specifies a time when a vulnerability assessment was last modified.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/modifiedTime`

Name:	modifiedTime
Nature:	DataProperty
Range:	/Core/DateTime

Referenced

- /Security/VulnAssessmentRelationship
- /Security/Vulnerability

10.2.12 percentile

Summary

The percentile of the current probability score.

Description

The percentile between 0 and 1 (0 and 100%) of the current probability score, the proportion of all scored vulnerabilities with the same or a lower probability score. The definition follows “percentile” in EPSS Data[77].

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/percentile`

Name:	percentile
Nature:	DataProperty
Range:	xsd:decimal

Referenced

- /Security/EpssVulnAssessmentRelationship

10.2.13 probability

Summary

A probability score between 0 and 1 of a vulnerability being exploited.

[77] https://www.first.org/epss/data_stats

Description

The probability score between 0 and 1 (0 and 100%) estimating the likelihood of exploitation in the wild in the next 30 days (following score publication). The definition follows “epss” in EPSS Data[78].

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/probability`

Name:	probability
Nature:	DataProperty
Range:	xsd:decimal

Referenced

- /Security/EpssVulnAssessmentRelationship

10.2.14 publishedTime

Summary

Specifies the time when a vulnerability was published.

Description

Specifies the time when a vulnerability was first published.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/publishedTime`

Name:	publishedTime
Nature:	DataProperty
Range:	/Core/DateTime

Referenced

- /Security/VulnAssessmentRelationship
- /Security/Vulnerability

10.2.15 score

Summary

Provides a numerical (0-10) representation of the severity of a vulnerability.

Description

The score provides information on the severity of a vulnerability per the Common Vulnerability Scoring System as defined by Forum of Incident Response and Security Teams[79].

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/score`

Name:	score
Nature:	DataProperty
Range:	xsd:decimal

Referenced

- /Security/CvssV2VulnAssessmentRelationship
- /Security/CvssV3VulnAssessmentRelationship
- /Security/CvssV4VulnAssessmentRelationship

[78] https://www.first.org/epss/data_stats

[79] https://www.first.org/cvss/

10.2.16 severity

Summary

Specifies the CVSS qualitative severity rating of a vulnerability in relation to a piece of software.

Description

The severity field provides a human readable string of the resulting numerical CVSS score.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/severity`

Name:	severity
Nature:	ObjectProperty
Range:	CvssSeverityType

Referenced

- /Security/CvssV3VulnAssessmentRelationship
- /Security/CvssV4VulnAssessmentRelationship

10.2.17 statusNotes

Summary

Conveys information about how VEX status was determined.

Description

A VEX statement may convey information about how status was determined and may reference other VEX information.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/statusNotes`

Name:	statusNotes
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Security/VexVulnAssessmentRelationship

10.2.18 vectorString

Summary

Specifies the CVSS vector string for a vulnerability.

Description

Specifies any combination of the CVSS Base, Temporal, Threat, Environmental, and/or Supplemental vector string values for a vulnerability.

Supports vectorStrings specified in all CVSS versions.

Constraints

String values for the vectorString range must only include the abbreviated form of metric names specified in CVSS specifications, e.g. Common Vulnerability Scoring System Vector String[80].

[80] https://www.first.org/cvss/v4.0/specification-document#Vector-String

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/vectorString`

Name:	vectorString
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Security/CvssV2VulnAssessmentRelationship
- /Security/CvssV3VulnAssessmentRelationship
- /Security/CvssV4VulnAssessmentRelationship

10.2.19 vexVersion

Summary

Specifies the version of a VEX statement.

Description

The statement version default value is zero. When any VEX-related content changes, the version must be incremented.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/vexVersion`

Name:	vexVersion
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Security/VexVulnAssessmentRelationship

10.2.20 withdrawnTime

Summary

Specified the time and date when a vulnerability was withdrawn.

Description

Specified the time and date when a vulnerability was withdrawn.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/withdrawnTime`

Name:	withdrawnTime
Nature:	DataProperty
Range:	/Core/DateTime

Referenced

- /Security/VulnAssessmentRelationship
- /Security/Vulnerability

10.3 Vocabularies

10.3.1 CvssSeverityType

Summary

Specifies the CVSS base, temporal, threat, or environmental severity type.

Description

CvssSeverityType specifies the Common Vulnerability Scoring System (CVSS) severity type, defined in the CVSS specifications as the textual representation of the numeric CVSS score.

The severity type entries are inclusive of and applicable to enumerations found in Common Vulnerability Scoring System v3.0: Specification Document[81] and Common Vulnerability Scoring System version 4.0: Specification Document[82].

CvssSeverityType is a mandatory field because baseSeverity is required in the CVSS 3.0 schema[83], CVSS 3.1 schema[84], and CVSS 4.0 schema[85].

The field can be used to document the base, temporal, threat, or environmental severity.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/CvssSeverityType`

Name:	CvssSeverityType

Entries

critical When a CVSS score is between 9.0 - 10.0

high When a CVSS score is between 7.0 - 8.9

low When a CVSS score is between 0.1 - 3.9

medium When a CVSS score is between 4.0 - 6.9

none When a CVSS score is 0.0

10.3.2 ExploitCatalogType

Summary

Specifies the exploit catalog type.

Description

ExploitCatalogType specifies the type of exploit catalog that a vulnerability is listed in.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/ExploitCatalogType`

Name:	ExploitCatalogType

Entries

kev CISA's Known Exploited Vulnerability (KEV) Catalog

other Other exploit catalogs

10.3.3 SsvcDecisionType

Summary

Specifies the SSVC decision type.

Description

SsvcDecisionType specifies the type of decision that's been made according to the Stakeholder-Specific Vulnerability Categorization (SSVC)[86].

[81] https://www.first.org/cvss/v3.0/specification-document#Qualitative-Severity-Rating-Scale

[82] https://www.first.org/cvss/v4.0/specification-document#Qualitative-Severity-Rating-Scale

[83] https://www.first.org/cvss/cvss-v3.0.json

[84] https://www.first.org/cvss/cvss-v3.1.json

[85] https://www.first.org/cvss/cvss-v4.0.json

[86] https://www.cisa.gov/stakeholder-specific-vulnerability-categorization-ssvc

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/SsvcDecisionType`

Name:	SsvcDecisionType

Entries

act The vulnerability requires attention from the organization's internal, supervisory-level and leadership-level individuals. Necessary actions include requesting assistance or information about the vulnerability, as well as publishing a notification either internally and/or externally. Typically, internal groups would meet to determine the overall response and then execute agreed upon actions. CISA recommends remediating Act vulnerabilities as soon as possible.

attend The vulnerability requires attention from the organization's internal, supervisory-level individuals. Necessary actions include requesting assistance or information about the vulnerability, and may involve publishing a notification either internally and/or externally. CISA recommends remediating Attend vulnerabilities sooner than standard update timelines.

track The vulnerability does not require action at this time. The organization would continue to track the vulnerability and reassess it if new information becomes available. CISA recommends remediating Track vulnerabilities within standard update timelines.

trackStar ("Track*" in the SSVC spec) The vulnerability contains specific characteristics that may require closer monitoring for changes. CISA recommends remediating Track* vulnerabilities within standard update timelines.

10.3.4 VexJustificationType

Summary

Specifies the VEX justification type.

Description

VexJustificationType specifies the type of Vulnerability Exploitability eXchange (VEX) justification.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Security/VexJustificationType`

Name:	VexJustificationType

Entries

componentNotPresent The software is not affected because the vulnerable component is not in the product.

inlineMitigationsAlreadyExist Built-in inline controls or mitigations prevent an adversary from leveraging the vulnerability.

vulnerableCodeCannotBeControlledByAdversary The vulnerable component is present, and the component contains the vulnerable code. However, vulnerable code is used in such a way that an attacker cannot mount any anticipated attack.

vulnerableCodeNotInExecutePath The affected code is not reachable through the execution of the code, including non-anticipated states of the product.

vulnerableCodeNotPresent The product is not affected because the code underlying the vulnerability is not present in the product.

11 Licensing

Summary

The Licensing Profile defines a minimum set of license information to facilitate compliance with typical license use cases.

Description

The Licensing profile only contains the additional requirement that any Software Artifact must have a `Relationship` of type `hasConcludedLicense`.

Classes and Property restrictions are defined in the SimpleLicensing Profile (Classes and Properties associated with license expression strings[87]) and in the ExpandedLicensing Profile (Classes and Properties used for a fully parsed syntax tree of license expressions).

There are 2 relationship types related to licensing - `hasDeclaredLicense` and `hasConcludedLicense`.

If the `hasConcludedLicense` for a Software Artifact is not the same as its `hasDeclaredLicense`, a written explanation SHOULD be provided in the `hasConcludedLicense` relationship `comment` field.

A written explanation of a relationship to a `NoAssertionLicense` MAY be provided in the `comment` field for the relationship.

hasDeclaredLicense

A hasDeclaredLicense identifies the license information actually found in the Software Artifact, for example as detected by use of automated tooling.

This field is not intended to capture license information obtained from an external source, such as a package's website. Such information can be included, as needed, in the hasConcludedLicense field.

A hasDeclaredLicense may be expressed differently in practice for different types of Software Artifacts. For example:

- for Packages, it would include license info for the Package as a whole, found in the Package itself (e.g., LICENSE file, README file, metadata in the Package, etc.), but it would not include any license information that is not in the Package itself (e.g., license information from the project's website or from a third party repository or website).
- for Files, it would include license info found in the File itself (e.g., license header or notice, comments indicating the license, SPDX-License-Identifier expression), but it would not include license info found in a different file (e.g., LICENSE file in the top directory of a repository).
- for Snippets, it would include license info found in the Snippet itself (e.g., license notice, comments, SPDX-License-Identifier expression), but it would not include license info found elsewhere in the File or in a different File (e.g., comment at top of File if it is not within the Snippet, LICENSE file in the top directory of a repository).

A hasDeclaredLicense relationship to NoneLicense indicates that the corresponding Package, File or Snippet contains no license information whatsoever.

A hasDeclaredLicense relationship to NoAssertionLicense indicates that one of the following applies:

- the SPDX data creator has attempted to but cannot reach a reasonable objective determination;
- the SPDX data creator has made no attempt to determine this field; or
- the SPDX data creator has intentionally provided no information (no meaning should be implied by doing so).

If a hasDeclaredLicense relationship is not present, no assumptions can be made about whether or not a hasDeclaredLicense exists.

Note that a missing hasDeclaredLicense is not the same as a relationship to NoAssertionLicense since the latter is a "known unknown" whereas no assumptions can be made from a missing hasDeclaredLicense relationship.

hasConcludedLicense

A hasConcludedLicense is the license identified by the SPDX data creator, based on analyzing the license information in the Software Artifact and other information to arrive at a reasonably objective conclusion as to what license governs the Software Artifact.

A hasConcludedLicense relationship to NoneLicense indicates that the SPDX data creator has looked and did not find any license information for this Software Artifact.

A hasConcludedLicense relationship to NoAssertionLicense indicates that one of the following applies:

[87] ../../annexes/spdx-license-expressions.md

- the SPDX data creator has attempted to but cannot reach a reasonable objective determination;
- the SPDX data creator has made no attempt to determine this field; or
- the SPDX data creator has intentionally provided no information (no meaning should be implied by doing so).

If a hasConcludedLicense is not present, no assumptions can be made about whether or not a hasConcludedLicense exists.

Note that a missing hasConcludedLicense is not the same as a relationship to a NoAssertionLicense since the latter is a “known unknown” whereas no assumptions can be made from a missing hasConcludedLicense relationship.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Licensing`

Name:	Licensing

Profile conformance

For an element collection to be conformant with this profile, the following has to hold:

1. for every `/Software/SoftwareArtifact` there MUST exist exactly one `/Core/Relationship` of type `hasConcludedLicense` having that element as its `from` property and a `/SimpleLicensing/AnyLicenseInfo` as its `to` property.

12 SimpleLicensing

Summary

Additional metadata relating to software licensing.

Description

The SimpleLicensing profile provides classes and properties to express licenses as a license expression[88] string.

It also provides the base abstract class, AnyLicenseInfo, used for references to license information.

The SimpleLicensingText class provides a place to record any license text found that does not match a license on the SPDX License List[89].

The ExpandedLicensing profile can be used to represent the complete parsed license expressions.

Metadata

```
https://spdx.org/rdf/3.0.1/terms/SimpleLicensing
```

Name:	SimpleLicensing

12.1 Classes

12.1.1 AnyLicenseInfo

Summary

Abstract class representing a license combination consisting of one or more licenses.

Description

AnyLicenseInfo is an abstract class representing a license combination consisting of one or more licenses (optionally including additional text), which may be combined according to the SPDX license expression syntax[90].

An AnyLicenseInfo is used by licensing properties of software artifacts.

It can be:

- a NoneLicense;
- a NoAssertionLicense;
- a single license (either on the SPDX License List[91] or a custom-defined license[92]);
- a single license with an “or later” operator applied;
- the foregoing with additional text applied; or
- a set of licenses combined by applying “AND” and “OR” operators recursively.

Metadata

```
https://spdx.org/rdf/3.0.1/terms/SimpleLicensing/AnyLicenseInfo
```

Name:	AnyLicenseInfo
Instantiability:	Abstract
SubclassOf:	/Core/Element

Superclasses

- /Core/Element

[88] ../../annexes/spdx-license-expressions.md

[89] https://spdx.org/licenses/

[90] ../../../annexes/spdx-license-expressions.md

[91] https://spdx.org/licenses/

[92] ../../ExpandedLicensing/Classes/CustomLicense.md

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
name	xsd:string	0	1
spdxId	xsd:anyURI	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

12.1.2 LicenseExpression

Summary

An SPDX Element containing an SPDX license expression string.

Description

A LicenseExpression enables the representation, in a single string, of a combination of one or more licenses, together with additions such as license exceptions.

The syntax for a licenseExpression string is set forth in the corresponding Annex of this specification ("SPDX license expressions"[93]). A licenseExpression string is not valid if it does not conform to the grammar set forth in that Annex.

The ExpandedLicensing profile can be used to represent the complete parsed license expression as a combination of license objects.

Metadata

`https://spdx.org/rdf/3.0.1/terms/SimpleLicensing/LicenseExpression`

Name:	LicenseExpression
Instantiability:	Concrete
SubclassOf:	AnyLicenseInfo

Superclasses

- /SimpleLicensing/AnyLicenseInfo
- /Core/Element

Properties

Property	Type	minCount	maxCount
customIdToUri	/Core/DictionaryEntry	0	*
licenseExpression	xsd:string	1	1
licenseListVersion	/Core/SemVer	0	1

[93] ../../../annexes/spdx-license-expressions.md

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
customIdToUri	/Core/DictionaryEntry	0	*
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
licenseExpression	xsd:string	1	1
licenseListVersion	/Core/SemVer	0	1
name	xsd:string	0	1
spdxId	xsd:anyURI	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

12.1.3 SimpleLicensingText

Summary

A license or addition that is not listed on the SPDX License List.

Description

A SimpleLicensingText represents a License or Addition that is not listed on the SPDX License List[94], and is therefore defined by an SPDX data creator.

Metadata

`https://spdx.org/rdf/3.0.1/terms/SimpleLicensing/SimpleLicensingText`

Name:	SimpleLicensingText
Instantiability:	Concrete
SubclassOf:	/Core/Element

Superclasses

- /Core/Element

Properties

Property	Type	minCount	maxCount
licenseText	xsd:string	1	1

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
licenseText	xsd:string	1	1
name	xsd:string	0	1
spdxId	xsd:anyURI	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

[94] https://spdx.org/licenses

12.2 Properties

12.2.1 customIdToUri

Summary

Maps a LicenseRef or AdditionRef string for a Custom License or a Custom License Addition to its URI ID.

Description

Within a License Expression, references can be made to a Custom License or a Custom License Addition.

The License Expression syntax[95] dictates any reference starting with a "LicenseRef-" or "AdditionRef-" refers to license or addition text not found in the official SPDX License List[96].

These custom licenses must be a CustomLicense, a CustomLicenseAddition, or a SimpleLicensingText which are identified with a unique URI identifier.

The key for the DictionaryEntry is the string used in the license expression and the value is the URI for the corresponding CustomLicense, CustomLicenseAddition, or SimpleLicensingText.

Metadata

`https://spdx.org/rdf/3.0.1/terms/SimpleLicensing/customIdToUri`

Name:	customIdToUri
Nature:	ObjectProperty
Range:	/Core/DictionaryEntry

Referenced

- /SimpleLicensing/LicenseExpression

12.2.2 licenseExpression

Summary

A string in the license expression format.

Description

A licenseExpression enables the representation, in a single string, of a combination of one or more licenses, together with additions such as license exceptions.

The syntax for a licenseExpression string is set forth in the corresponding Annex of this specification ("SPDX license expressions"[97]). A licenseExpression string is not valid if it does not conform to the grammar set forth in that Annex.

The ExpandedLicensing profile can be used to represent the complete parsed license expression as a combination of license objects.

Metadata

`https://spdx.org/rdf/3.0.1/terms/SimpleLicensing/licenseExpression`

Name:	licenseExpression
Nature:	DataProperty
Range:	xsd:string

Referenced

- /SimpleLicensing/LicenseExpression

12.2.3 licenseListVersion

Summary

The version of the SPDX License List used in the license expression.

[95] ../../../annexes/spdx-license-expressions.md

[96] https://spdx.org/licenses/

[97] ../../../annexes/spdx-license-expressions.md

Description

Recognizing that licenses are added to the SPDX License List[98] with each subsequent version, the intent is to provide consumers with the version of the SPDX License List used.

This anticipates that in the future, license expression might have used a version of the SPDX License List that is older than the then current one.

The specified version of the SPDX License List must include all listed licenses and exceptions referenced in the expression.

Metadata

`https://spdx.org/rdf/3.0.1/terms/SimpleLicensing/licenseListVersion`

Name:	licenseListVersion
Nature:	DataProperty
Range:	/Core/SemVer

Referenced

- /SimpleLicensing/LicenseExpression

12.2.4 licenseText

Summary

Identifies the full text of a License or Addition.

Description

A licenseText contains the plain text of the License or Addition, without templating or other similar markup.

Users of the licenseText for a License can apply the SPDX License List Matching Guidelines[99] when comparing it to another text for matching purposes.

Metadata

`https://spdx.org/rdf/3.0.1/terms/SimpleLicensing/licenseText`

Name:	licenseText
Nature:	DataProperty
Range:	xsd:string

Referenced

- /ExpandedLicensing/License
- /SimpleLicensing/SimpleLicensingText

[98] https://spdx.org/licenses/

[99] ../../../annexes/license-matching-guidelines-and-templates.md

This page intentionally left blank.

13 ExpandedLicensing

Summary

Fully expanded license expressions.

Description

This profile supports representing a fully expanded license expression[100] in object form.

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing`

Name:	ExpandedLicensing

13.1 Classes

13.1.1 ConjunctiveLicenseSet

Summary

Portion of an AnyLicenseInfo representing a set of licensing information where all elements apply.

Description

A ConjunctiveLicenseSet indicates that *each* of its subsidiary AnyLicenseInfos apply. In other words, a ConjunctiveLicenseSet of two or more licenses represents a licensing situation where *all* of the specified licenses are to be complied with. It is represented in the SPDX License Expression Syntax by the `AND` operator.

It is syntactically correct to specify a ConjunctiveLicenseSet where the subsidiary AnyLicenseInfos may be "incompatible" according to a particular interpretation of the corresponding Licenses. The SPDX License Expression Syntax[101] does not take into account interpretation of license texts, which is left to the consumer of SPDX data to determine for themselves.

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/ConjunctiveLicenseSet`

Name:	ConjunctiveLicenseSet
Instantiability:	Concrete
SubclassOf:	/SimpleLicensing/AnyLicenseInfo

Superclasses

- /SimpleLicensing/AnyLicenseInfo
- /Core/Element

Properties

Property	Type	minCount	maxCount
member	/SimpleLicensing/AnyLicenseInfo	2	*

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
member	/SimpleLicensing/AnyLicenseInfo	2	*
name	xsd:string	0	1
spdxId	xsd:anyURI	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

[100] ../../annexes/spdx-license-expressions.md

[101] ../../../annexes/spdx-license-expressions.md

13.1.2 CustomLicense

Summary

A license that is not listed on the SPDX License List.

Description

A CustomLicense represents a License that is not listed on the SPDX License List[102], and is therefore defined by an SPDX data creator.

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/CustomLicense`

Name:	CustomLicense
Instantiability:	Concrete
SubclassOf:	License

Superclasses

- /ExpandedLicensing/License
- /ExpandedLicensing/ExtendableLicense
- /SimpleLicensing/AnyLicenseInfo
- /Core/Element

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
isDeprecatedLicenseId	xsd:boolean	0	1
isFsfLibre	xsd:boolean	0	1
isOsiApproved	xsd:boolean	0	1
licenseText	xsd:string	1	1
licenseXml	xsd:string	0	1
name	xsd:string	0	1
obsoletedBy	xsd:string	0	1
seeAlso	xsd:anyURI	0	*
spdxId	xsd:anyURI	1	1
standardLicenseHeader	xsd:string	0	1
standardLicenseTemplate	xsd:string	0	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

13.1.3 CustomLicenseAddition

Summary

A license addition that is not listed on the SPDX Exceptions List.

Description

A CustomLicenseAddition represents an addition to a License that is not listed on the SPDX License Exceptions[103], and is therefore defined by an SPDX data creator.

It is intended to represent additional language which is meant to be added to a License, but which is not itself a standalone License.

[102] https://spdx.org/licenses

[103] https://spdx.org/licenses/exceptions-index.html

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/CustomLicenseAddition`

Name:	CustomLicenseAddition
Instantiability:	Concrete
SubclassOf:	LicenseAddition

Superclasses

- /ExpandedLicensing/LicenseAddition
- /Core/Element

All properties (informative)

Property	Type	minCount	maxCount
additionText	xsd:string	1	1
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
isDeprecatedAdditionId	xsd:boolean	0	1
licenseXml	xsd:string	0	1
name	xsd:string	0	1
obsoletedBy	xsd:string	0	1
seeAlso	xsd:anyURI	0	*
spdxId	xsd:anyURI	1	1
standardAdditionTemplate	xsd:string	0	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

13.1.4 DisjunctiveLicenseSet

Summary

Portion of an AnyLicenseInfo representing a set of licensing information where only one of the elements applies.

Description

A DisjunctiveLicenseSet indicates that *only one* of its subsidiary AnyLicenseInfos is required to apply. In other words, a DisjunctiveLicenseSet of two or more licenses represents a licensing situation where *only one* of the specified licenses are to be complied with.

A consumer of SPDX data would typically understand this to permit the recipient of the licensed content to choose which of the corresponding license they would prefer to use. It is represented in the SPDX License Expression Syntax by the `OR` operator.

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/DisjunctiveLicenseSet`

Name:	DisjunctiveLicenseSet
Instantiability:	Concrete
SubclassOf:	/SimpleLicensing/AnyLicenseInfo

Superclasses

- /SimpleLicensing/AnyLicenseInfo
- /Core/Element

Properties

Property	Type	minCount	maxCount
member	/SimpleLicensing/AnyLicenseInfo	2	*

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
member	/SimpleLicensing/AnyLicenseInfo	2	*
name	xsd:string	0	1
spdxId	xsd:anyURI	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

13.1.5 ExtendableLicense

Summary

Abstract class representing a License or an OrLaterOperator.

Description

The WithAdditionOperator can have a License or an OrLaterOperator as the license property value. This class is used for the value.

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/ExtendableLicense`

Name:	ExtendableLicense
Instantiability:	Abstract
SubclassOf:	/SimpleLicensing/AnyLicenseInfo

Superclasses

- /SimpleLicensing/AnyLicenseInfo
- /Core/Element

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
name	xsd:string	0	1
spdxId	xsd:anyURI	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

13.1.6 IndividualLicensingInfo

Summary

A concrete subclass of AnyLicenseInfo used by Individuals in the ExpandedLicensing profile.

Description

Individuals, such as NoneLicense and NoAssertionLicense, need to reference a concrete subclass of AnyLicenseInfo.

This class provides the type used by the individuals.

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/IndividualLicensingInfo`

Name:	IndividualLicensingInfo
Instantiability:	Concrete
SubclassOf:	/SimpleLicensing/AnyLicenseInfo

Superclasses

- /SimpleLicensing/AnyLicenseInfo
- /Core/Element

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
name	xsd:string	0	1
spdxId	xsd:anyURI	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

13.1.7 License

Summary

Abstract class for the portion of an AnyLicenseInfo representing a license.

Description

A License represents a license text, whether listed on the SPDX License List[104] (ListedLicense) or defined by an SPDX data creator (CustomLicense).

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/License`

Name:	License
Instantiability:	Abstract
SubclassOf:	ExtendableLicense

Superclasses

- /ExpandedLicensing/ExtendableLicense
- /SimpleLicensing/AnyLicenseInfo
- /Core/Element

Properties

Property	Type	minCount	maxCount
/SimpleLicensing/licenseText	xsd:string	1	1
isDeprecatedLicenseId	xsd:boolean	0	1
isFsfLibre	xsd:boolean	0	1
isOsiApproved	xsd:boolean	0	1
licenseXml	xsd:string	0	1
obsoletedBy	xsd:string	0	1
seeAlso	xsd:anyURI	0	*
standardLicenseHeader	xsd:string	0	1
standardLicenseTemplate	xsd:string	0	1

[104] https://spdx.org/licenses/

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
isDeprecatedLicenseId	xsd:boolean	0	1
isFsfLibre	xsd:boolean	0	1
isOsiApproved	xsd:boolean	0	1
licenseText	xsd:string	1	1
licenseXml	xsd:string	0	1
name	xsd:string	0	1
obsoletedBy	xsd:string	0	1
seeAlso	xsd:anyURI	0	*
spdxId	xsd:anyURI	1	1
standardLicenseHeader	xsd:string	0	1
standardLicenseTemplate	xsd:string	0	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

13.1.8 LicenseAddition

Summary

Abstract class for additional text intended to be added to a License, but which is not itself a standalone License.

Description

A LicenseAddition represents text which is intended to be added to a License as additional text, but which is not itself intended to be a standalone License.

It may be an exception which is listed on the SPDX License Exceptions[105] (ListedLicenseException), or may be any other additional text (as an exception or otherwise) which is defined by an SPDX data creator (CustomLicenseAddition).

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/LicenseAddition`

Name:	LicenseAddition
Instantiability:	Abstract
SubclassOf:	/Core/Element

Superclasses

- /Core/Element

Properties

Property	Type	minCount	maxCount
additionText	xsd:string	1	1
isDeprecatedAdditionId	xsd:boolean	0	1
licenseXml	xsd:string	0	1
obsoletedBy	xsd:string	0	1
seeAlso	xsd:anyURI	0	*
standardAdditionTemplate	xsd:string	0	1

[105] https://spdx.org/licenses/exceptions-index.html

All properties (informative)

Property	Type	minCount	maxCount
additionText	xsd:string	1	1
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
isDeprecatedAdditionId	xsd:boolean	0	1
licenseXml	xsd:string	0	1
name	xsd:string	0	1
obsoletedBy	xsd:string	0	1
seeAlso	xsd:anyURI	0	*
spdxId	xsd:anyURI	1	1
standardAdditionTemplate	xsd:string	0	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

13.1.9 ListedLicense

Summary

A license that is listed on the SPDX License List.

Description

A ListedLicense represents a License that is listed on the SPDX License List[106].

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/ListedLicense`

Name:	ListedLicense
Instantiability:	Concrete
SubclassOf:	License

Superclasses

- /ExpandedLicensing/License
- /ExpandedLicensing/ExtendableLicense
- /SimpleLicensing/AnyLicenseInfo
- /Core/Element

Properties

Property	Type	minCount	maxCount
deprecatedVersion	xsd:string	0	1
listVersionAdded	xsd:string	0	1

[106] https://spdx.org/licenses

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
deprecatedVersion	xsd:string	0	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
isDeprecatedLicenseId	xsd:boolean	0	1
isFsfLibre	xsd:boolean	0	1
isOsiApproved	xsd:boolean	0	1
licenseText	xsd:string	1	1
licenseXml	xsd:string	0	1
listVersionAdded	xsd:string	0	1
name	xsd:string	0	1
obsoletedBy	xsd:string	0	1
seeAlso	xsd:anyURI	0	*
spdxId	xsd:anyURI	1	1
standardLicenseHeader	xsd:string	0	1
standardLicenseTemplate	xsd:string	0	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

13.1.10 ListedLicenseException

Summary

A license exception that is listed on the SPDX Exceptions list.

Description

A ListedLicenseException represents an exception to a License (in other words, an exception to a license condition or an additional permission beyond those granted in a License) which is listed on the SPDX License Exceptions[107].

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/ListedLicenseException`

Name:	ListedLicenseException
Instantiability:	Concrete
SubclassOf:	LicenseAddition

Superclasses

- /ExpandedLicensing/LicenseAddition
- /Core/Element

Properties

Property	Type	minCount	maxCount
deprecatedVersion	xsd:string	0	1
listVersionAdded	xsd:string	0	1

[107] https://spdx.org/licenses/exceptions-index.html

All properties (informative)

Property	Type	minCount	maxCount
additionText	xsd:string	1	1
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
deprecatedVersion	xsd:string	0	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
isDeprecatedAdditionId	xsd:boolean	0	1
licenseXml	xsd:string	0	1
listVersionAdded	xsd:string	0	1
name	xsd:string	0	1
obsoletedBy	xsd:string	0	1
seeAlso	xsd:anyURI	0	*
spdxId	xsd:anyURI	1	1
standardAdditionTemplate	xsd:string	0	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

13.1.11 OrLaterOperator

Summary

Portion of an AnyLicenseInfo representing this version, or any later version, of the indicated License.

Description

An OrLaterOperator indicates that this portion of the AnyLicenseInfo represents either (1) the specified version of the corresponding License, or (2) any later version of that License. It is represented in the SPDX License Expression Syntax by the + operator.

It is context-dependent, and unspecified by SPDX, as to what constitutes a "later version" of any particular License. Some Licenses may not be versioned, or may not have clearly-defined ordering for versions. The consumer of SPDX data will need to determine for themselves what meaning to attribute to a "later version" operator for a particular License.

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/OrLaterOperator`

Name:	OrLaterOperator
Instantiability:	Concrete
SubclassOf:	ExtendableLicense

Superclasses

- /ExpandedLicensing/ExtendableLicense
- /SimpleLicensing/AnyLicenseInfo
- /Core/Element

Properties

Property	Type	minCount	maxCount
subjectLicense	License	1	1

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
name	xsd:string	0	1
spdxId	xsd:anyURI	1	1
subjectLicense	License	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

13.1.12 WithAdditionOperator

Summary

Portion of an AnyLicenseInfo representing a License which has additional text applied to it.

Description

A WithAdditionOperator indicates that the designated License is subject to the designated LicenseAddition, which might be a license exception on the SPDX License Exceptions[108] (ListedLicenseException) or may be other additional text (CustomLicenseAddition). It is represented in the SPDX License Expression Syntax by the `WITH` operator.

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/WithAdditionOperator`

Name:	WithAdditionOperator
Instantiability:	Concrete
SubclassOf:	/SimpleLicensing/AnyLicenseInfo

Superclasses

- /SimpleLicensing/AnyLicenseInfo
- /Core/Element

Properties

Property	Type	minCount	maxCount
subjectAddition	LicenseAddition	1	1
subjectExtendableLicense	ExtendableLicense	1	1

All properties (informative)

Property	Type	minCount	maxCount
comment	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
name	xsd:string	0	1
spdxId	xsd:anyURI	1	1
subjectAddition	LicenseAddition	1	1
subjectExtendableLicense	ExtendableLicense	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

[108] https://spdx.org/licenses/exceptions-index.html

13.2 Properties

13.2.1 additionText

Summary

Identifies the full text of a LicenseAddition.

Description

An additionText contains the plain text of the LicenseAddition, without templating or other similar markup.

Users of the additionText for a License can apply the SPDX License List Matching Guidelines[109] when comparing it to another text for matching purposes.

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/additionText`

Name:	additionText
Nature:	DataProperty
Range:	xsd:string

Referenced

- /ExpandedLicensing/LicenseAddition

13.2.2 deprecatedVersion

Summary

Specifies the SPDX License List version in which this license or exception identifier was deprecated.

Description

A deprecatedVersion, for a ListedLicense on the SPDX License List[110] or a ListedLicenseException on the SPDX License Exceptions[111], specifies which version release of the License List was the first one in which it was marked as deprecated.

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/deprecatedVersion`

Name:	deprecatedVersion
Nature:	DataProperty
Range:	xsd:string

Referenced

- /ExpandedLicensing/ListedLicense
- /ExpandedLicensing/ListedLicenseException

13.2.3 isDeprecatedAdditionId

Summary

Specifies whether an additional text identifier has been marked as deprecated.

Description

The isDeprecatedAdditionId property specifies whether an identifier for a LicenseAddition has been marked as deprecated. If the property is not defined, then it is presumed to be false (i.e., not deprecated).

If the LicenseAddition is included on the SPDX License Exceptions[112], then the `deprecatedVersion` property indicates on which version release of the Exceptions List it was first marked as deprecated.

[109] ../../../annexes/license-matching-guidelines-and-templates.md

[110] https://spdx.org/licenses/

[111] https://spdx.org/licenses/exceptions-index.html

[112] https://spdx.org/licenses/exceptions-index.html

"Deprecated" in this context refers to deprecating the use of the *identifier*, not the underlying license addition. In other words, even if a LicenseAddition's author or steward has stated that a particular LicenseAddition generally should not be used, that would *not* mean that the LicenseAddition's identifier is "deprecated." Rather, a LicenseAddition operator is typically marked as "deprecated" when it is determined that use of another identifier is preferable.

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/isDeprecatedAdditionId`

Name:	isDeprecatedAdditionId
Nature:	DataProperty
Range:	xsd:boolean

Referenced

- /ExpandedLicensing/LicenseAddition

13.2.4 isDeprecatedLicenseId

Summary

Specifies whether a license or additional text identifier has been marked as deprecated.

Description

The isDeprecatedLicenseId property specifies whether an identifier for a License or LicenseAddition has been marked as deprecated. If the property is not defined, then it is presumed to be false (i.e., not deprecated).

If the License or LicenseAddition is included on the SPDX License List[113], then the `deprecatedVersion` property indicates on which version release of the License List it was first marked as deprecated.

"Deprecated" in this context refers to deprecating the use of the *identifier*, not the underlying license. In other words, even if a License's author or steward has stated that a particular License generally should not be used, that would *not* mean that the License's identifier is "deprecated." Rather, a License or LicenseAddition operator is typically marked as "deprecated" when it is determined that use of another identifier is preferable.

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/isDeprecatedLicenseId`

Name:	isDeprecatedLicenseId
Nature:	DataProperty
Range:	xsd:boolean

Referenced

- /ExpandedLicensing/License

13.2.5 isFsfLibre

Summary

Specifies whether the License is listed as free by the Free Software Foundation (FSF).

Description

isFsfLibre specifies whether the Free Software Foundation (FSF)[114] has listed this License as "free" in their commentary on licenses, located at the time of this writing at Various Licenses and Comments about Them[115].

A value of "true" indicates that the license is in the list of licenses that FSF publishes as libre.

A value of "false" indicates that the license is explicitly not in the corresponding list of FSF libre licenses (e.g., FSF has the license on a non-free list).

If the isFsfLibre field is not specified, the SPDX data creator makes no assertions about whether the License is listed in the FSF's commentary.

[113] https://spdx.org/licenses/

[114] https://fsf.org

[115] https://www.gnu.org/licenses/license-list.en.html

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/isFsfLibre`

Name:	isFsfLibre
Nature:	DataProperty
Range:	xsd:boolean

Referenced

- /ExpandedLicensing/License

13.2.6 isOsiApproved

Summary

Specifies whether the License is listed as approved by the Open Source Initiative (OSI).

Description

isOsiApproved specifies whether the Open Source Initiative (OSI)[116] has listed this License as "approved" in their list of OSI Approved Licenses, located at the time of this writing at OSI Approved Licenses[117].

A value of "true" indicates that the license is in the list of licenses that OSI publishes as approved.

A value of "false" indicates that the license is explicitly not in the corresponding list of OSI licenses (e.g., OSI has stated publicly that a license is not approved).

If the isOsiApproved field is not specified, the SPDX data creator makes no assertions about whether the License is approved by the OSI.

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/isOsiApproved`

Name:	isOsiApproved
Nature:	DataProperty
Range:	xsd:boolean

Referenced

- /ExpandedLicensing/License

13.2.7 licenseXml

Summary

Identifies all the text and metadata associated with a license in the license XML format.

Description

The license XML format is defined and used by the SPDX legal team.

The formal schema definition is available at SPDX License List XML Schema[118].

For a text description of the XML fields, see XML template fields[119].

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/licenseXml`

Name:	licenseXml
Nature:	DataProperty
Range:	xsd:string

[116] https://opensource.org

[117] https://opensource.org/licenses

[118] https://github.com/spdx/license-list-XML/blob/v3.25.0/schema/ListedLicense.xsd

[119] https://github.com/spdx/license-list-XML/blob/v3.25.0/DOCS/xml-fields.md

Referenced

- /ExpandedLicensing/License
- /ExpandedLicensing/LicenseAddition

13.2.8 listVersionAdded

Summary

Specifies the SPDX License List version in which this ListedLicense or ListedLicenseException identifier was first added.

Description

A listVersionAdded for a ListedLicense or ListedLicenseException on the SPDX License List[120] specifies which version release of the License List was the first one in which it was included.

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/listVersionAdded`

Name:	listVersionAdded
Nature:	DataProperty
Range:	xsd:string

Referenced

- /ExpandedLicensing/ListedLicense
- /ExpandedLicensing/ListedLicenseException

13.2.9 member

Summary

A license expression participating in a license set.

Description

A member is a license expression participating in a conjunctive (of type ConjunctiveLicenseSet) or a disjunctive (of type DisjunctiveLicenseSet) license set.

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/member`

Name:	member
Nature:	ObjectProperty
Range:	/SimpleLicensing/AnyLicenseInfo

Referenced

- /ExpandedLicensing/ConjunctiveLicenseSet
- /ExpandedLicensing/DisjunctiveLicenseSet

13.2.10 obsoletedBy

Summary

Specifies the licenseId that is preferred to be used in place of a deprecated License or LicenseAddition.

Description

An obsoletedBy value for a deprecated License or LicenseAddition specifies the licenseId of the replacement License or LicenseAddition that is preferred to be used in its place. It should use the same format as specified for a licenseId.

The License's or LicenseAddition's comment value may include more information about the reason why the licenseId specified in the obsoletedBy value is preferred.

[120] https://spdx.org/licenses/

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/obsoletedBy`

Name:	obsoletedBy
Nature:	DataProperty
Range:	xsd:string

Referenced

- /ExpandedLicensing/License
- /ExpandedLicensing/LicenseAddition

13.2.11 seeAlso

Summary

Contains a URL where the License or LicenseAddition can be found in use.

Description

A seeAlso defines a cross-reference with a URL where the License or LicenseAddition can be found in use by one or a few projects.

If applicable, it should include a URL where the license text is posted by the license steward, particularly if the license steward has made available a "canonical" primary URL for the license text.

If the license is OSI approved, a seeAlso should be included with the URL for the license's listing on the OSI website.

The seeAlso URL may refer to a previously-available URL for the License or LicenseAddition which is no longer active.

Where applicable, the seeAlso URL should include the license text in its native language. seeAlso URLs to English or other translations may be included where multiple, equivalent official translations exist.

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/seeAlso`

Name:	seeAlso
Nature:	DataProperty
Range:	xsd:anyURI

Referenced

- /ExpandedLicensing/License
- /ExpandedLicensing/LicenseAddition

13.2.12 standardAdditionTemplate

Summary

Identifies the full text of a LicenseAddition, in SPDX templating format.

Description

A standardAdditionTemplate contains a license addition template which describes sections of the LicenseAddition text which can be varied.

See the Legacy Text Template format section of the SPDX License List Matching Guidelines[121] for format information.

It is recommended to use licenseXml[122] instead, as it can capture all the text and metadata associated with a license.

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/standardAdditionTemplate`

Name:	standardAdditionTemplate
Nature:	DataProperty
Range:	xsd:string

[121] ../../../annexes/license-matching-guidelines-and-templates.md

[122] ./licenseXml.md

Referenced

- /ExpandedLicensing/LicenseAddition

13.2.13 standardLicenseHeader

Summary

Provides a License author's preferred text to indicate that a file is covered by the License.

Description

A standardLicenseHeader contains the plain text of the License author's preferred wording to be used, typically in a source code file's header comments or similar location, to indicate that the file is subject to the specified License.

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/standardLicenseHeader`

Name:	standardLicenseHeader
Nature:	DataProperty
Range:	xsd:string

Referenced

- /ExpandedLicensing/License

13.2.14 standardLicenseTemplate

Summary

Identifies the full text of a License, in SPDX templating format.

Description

A standardLicenseTemplate contains a license template which describes sections of the License text which can be varied.

See the Legacy Text Template format section of the SPDX License List Matching Guidelines[123] for format information.

It is recommended to use licenseXml[124] instead, as it can capture all the text and metadata associated with a license.

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/standardLicenseTemplate`

Name:	standardLicenseTemplate
Nature:	DataProperty
Range:	xsd:string

Referenced

- /ExpandedLicensing/License

13.2.15 subjectAddition

Summary

A LicenseAddition participating in a 'with addition' model.

Description

A subjectAddition is a LicenseAddition which is subject to a 'with additional text' effect (WithAdditionOperator).

[123] ../../../annexes/license-matching-guidelines-and-templates.md

[124] ./licenseXml.md

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/subjectAddition`

Name:	subjectAddition
Nature:	ObjectProperty
Range:	LicenseAddition

Referenced

- /ExpandedLicensing/WithAdditionOperator

13.2.16 subjectExtendableLicense

Summary

A License participating in a ’with addition’ model.

Description

A subjectExtendableLicense is a License which is subject to a ‘with additional text’ effect (WithAdditionOperator).

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/subjectExtendableLicense`

Name:	subjectExtendableLicense
Nature:	ObjectProperty
Range:	ExtendableLicense

Referenced

- /ExpandedLicensing/WithAdditionOperator

13.2.17 subjectLicense

Summary

A License participating in an ’or later’ model.

Description

A subjectLicense is a License which is subject an ‘or later’ effect (OrLaterOperator).

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/subjectLicense`

Name:	subjectLicense
Nature:	ObjectProperty
Range:	License

Referenced

- /ExpandedLicensing/OrLaterOperator

13.3 Individuals

13.3.1 NoAssertionLicense

Summary

An Individual Value for License when no assertion can be made about its actual value.

Description

NoAssertionLicense should be used if

- the SPDX creator has attempted to but cannot reach a reasonable objective determination;
- the SPDX creator has made no attempt to determine this field; or
- the SPDX creator has intentionally provided no information (no meaning should be implied by doing so).

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/NoAssertionLicense`

Name:	NoAssertionLicense
Type:	IndividualLicensingInfo
IRI:	`https://spdx.org/rdf/3.0.1/terms/Licensing/NoAssertion`

13.3.2 NoneLicense

Summary

An Individual Value for License where the SPDX data creator determines that no license is present.

Description

NoneLicense should be used if the SPDX creator determines there is no license available for this Artifact.

Metadata

`https://spdx.org/rdf/3.0.1/terms/ExpandedLicensing/NoneLicense`

Name:	NoneLicense
Type:	IndividualLicensingInfo
IRI:	`https://spdx.org/rdf/3.0.1/terms/Licensing/None`

14 Dataset

Summary

The Dataset Profile provides additional metadata, based on Software Profile, that is useful for datasets.

Description

The Dataset namespace defines concepts related to dataset, including its preparation process, its characteristics, and its access methods.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Dataset`

Name:	Dataset

Profile conformance

For an element collection to be conformant with this profile, the following has to hold:

1. for every `/Dataset/DatasetPackage` there MUST exist exactly one `/Core/Relationship` of type `hasConcludedLicense` having that element as its `from` property and a `/SimpleLicensing/AnyLicenseInfo` as its `to` property.
2. for every `/Dataset/DatasetPackage` there MUST exist exactly one `/Core/Relationship` of type `hasDeclaredLicense` having that element as its `from` property and a `/SimpleLicensing/AnyLicenseInfo` as its `to` property.

14.1 Classes

14.1.1 DatasetPackage

Summary

Specifies a data package and its associated information.

Description

Metadata information that can be added to a dataset that may be used in a software or to train/test an AI package.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Dataset/DatasetPackage`

Name:	DatasetPackage
Instantiability:	Concrete
SubclassOf:	/Software/Package

Superclasses

- /Software/Package
- /Software/SoftwareArtifact
- /Core/Artifact
- /Core/Element

Properties

Property	Type	minCount	maxCount
anonymizationMethodUsed	xsd:string	0	*
confidentialityLevel	ConfidentialityLevelType	0	1
dataCollectionProcess	xsd:string	0	1
dataPreprocessing	xsd:string	0	*
datasetAvailability	DatasetAvailabilityType	0	1
datasetNoise	xsd:string	0	1
datasetSize	xsd:nonNegativeInteger	0	1
datasetType	DatasetType	1	*
datasetUpdateMechanism	xsd:string	0	1
hasSensitivePersonalInformation	/Core/PresenceType	0	1
intendedUse	xsd:string	0	1
knownBias	xsd:string	0	*
sensor	/Core/DictionaryEntry	0	*

External properties cardinality updates

Property	minCount	maxCount
/Core/Artifact/builtTime	1	
/Core/Artifact/originatedBy	1	1
/Core/Artifact/releaseTime	1	
/Software/Package/downloadLocation	1	
/Software/SoftwareArtifact/primaryPurpose	1	

All properties (informative)

Property	Type	minCount	maxCount
additionalPurpose	SoftwarePurpose	0	*
anonymizationMethodUsed	xsd:string	0	*
attributionText	xsd:string	0	*
builtTime	DateTime	1	1
comment	xsd:string	0	1
confidentialityLevel	ConfidentialityLevelType	0	1
contentIdentifier	ContentIdentifier	0	*
copyrightText	xsd:string	0	1
creationInfo	CreationInfo	1	1
dataCollectionProcess	xsd:string	0	1
dataPreprocessing	xsd:string	0	*
datasetAvailability	DatasetAvailabilityType	0	1
datasetNoise	xsd:string	0	1
datasetSize	xsd:nonNegativeInteger	0	1
datasetType	DatasetType	1	*
datasetUpdateMechanism	xsd:string	0	1
description	xsd:string	0	1
downloadLocation	xsd:anyURI	1	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
hasSensitivePersonalInformation	/Core/PresenceType	0	1
homePage	xsd:anyURI	0	1
intendedUse	xsd:string	0	1
knownBias	xsd:string	0	*
name	xsd:string	1	1
originatedBy	Agent	1	1
packageUrl	xsd:anyURI	0	1
packageVersion	xsd:string	0	1
primaryPurpose	SoftwarePurpose	1	1
releaseTime	DateTime	1	1
sensor	/Core/DictionaryEntry	0	*
sourceInfo	xsd:string	0	1
spdxId	xsd:anyURI	1	1
standardName	xsd:string	0	*
summary	xsd:string	0	1
suppliedBy	Agent	0	1
supportLevel	SupportType	0	*
validUntilTime	DateTime	0	1
verifiedUsing	IntegrityMethod	0	*

14.2 Properties

14.2.1 anonymizationMethodUsed

Summary

Describes the anonymization methods used.

Description

A free-form text that describes the methods used to anonymize the dataset or fields in the dataset.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Dataset/anonymizationMethodUsed`

Name:	anonymizationMethodUsed
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Dataset/DatasetPackage

14.2.2 confidentialityLevel

Summary

Describes the confidentiality level of the data points contained in the dataset.

Description

Describes the levels of confidentiality of the data points contained in the dataset.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Dataset/confidentialityLevel`

Name:	confidentialityLevel
Nature:	ObjectProperty
Range:	ConfidentialityLevelType

Referenced

- /Dataset/DatasetPackage

14.2.3 dataCollectionProcess

Summary

Describes how the dataset was collected.

Description

A free-form text that describes how a dataset was collected.

Examples include the sources from which a dataset was scrapped and the interview protocol that was used for data collection.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Dataset/dataCollectionProcess`

Name:	dataCollectionProcess
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Dataset/DatasetPackage

14.2.4 dataPreprocessing

Summary

Describes the preprocessing steps that were applied to the raw data to create the given dataset.

Description

A free-form text that describes the various preprocessing steps that were applied to the raw data to create the dataset.

Examples include standardization, normalization, deduplication, tokenization, and removal of tokens.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Dataset/dataPreprocessing`

Name:	dataPreprocessing
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Dataset/DatasetPackage

14.2.5 datasetAvailability

Summary

The field describes the availability of a dataset.

Description

Some datasets are publicly available and can be downloaded directly. Others are only accessible behind a clickthrough, or after filling a registration form. This field will describe the dataset availability from that perspective.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Dataset/datasetAvailability`

Name:	datasetAvailability
Nature:	ObjectProperty
Range:	DatasetAvailabilityType

Referenced

- /Dataset/DatasetPackage

14.2.6 datasetNoise

Summary

Describes potentially noisy elements of the dataset.

Description

Describes what kinds of noises a dataset might encompass.

The free-form text specifies fields or samples that might be noisy.

Alternatively, it can also be used to describe various noises that could impact the whole dataset.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Dataset/datasetNoise`

Name:	datasetNoise
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Dataset/DatasetPackage

14.2.7 datasetSize

Summary

Captures the size of the dataset.

Description

Captures how large a dataset is.

The size is to be measured in bytes.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Dataset/datasetSize`

Name:	datasetSize
Nature:	DataProperty
Range:	xsd:nonNegativeInteger

Referenced

- /Dataset/DatasetPackage

14.2.8 datasetType

Summary

Describes the type of the given dataset.

Description

Describes the datatype contained in the dataset.

For example, a dataset can be an image dataset for computer vision applications, a text dataset such as the contents of a book or Wikipedia article, or sometimes a multimodal dataset that contains multiple types of data.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Dataset/datasetType`

Name:	datasetType
Nature:	ObjectProperty
Range:	DatasetType

Referenced

- /Dataset/DatasetPackage

14.2.9 datasetUpdateMechanism

Summary

Describes a mechanism to update the dataset.

Description

A free-form text that describes a mechanism to update the dataset.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Dataset/datasetUpdateMechanism`

Name:	datasetUpdateMechanism
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Dataset/DatasetPackage

14.2.10 hasSensitivePersonalInformation

Summary

Describes if any sensitive personal information is present in the dataset.

Description

Indicates the presence of sensitive personal data or information that allows drawing conclusions about a person’s identity.

Related: `useSensitivePersonalInformation` in `/AI/AIPackage`

Metadata

`https://spdx.org/rdf/3.0.1/terms/Dataset/hasSensitivePersonalInformation`

Name:	hasSensitivePersonalInformation
Nature:	ObjectProperty
Range:	/Core/PresenceType

Referenced

- /Dataset/DatasetPackage

14.2.11 intendedUse

Summary

Describes what the given dataset should be used for.

Description

A free-form text that describes what the given dataset should be used for.

Some datasets are collected to be used only for particular purposes.

For example, medical data collected from a specific demography might only be applicable for training machine learning models to make predictions for that demography. In such a case, the intendedUse field would capture this information. Similarly, if a dataset is collected for building a facial recognition model, the intendedUse field would specify that.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Dataset/intendedUse`

Name:	intendedUse
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Dataset/DatasetPackage

14.2.12 knownBias

Summary

Records the biases that the dataset is known to encompass.

Description

A free-form text that describes the different biases that the dataset encompasses.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Dataset/knownBias`

Name:	knownBias
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Dataset/DatasetPackage

14.2.13 sensor

Summary

Describes a sensor used for collecting the data.

Description

Describes a sensor that was used for collecting the data and its calibration value as a key-value pair.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Dataset/sensor`

Name:	sensor
Nature:	ObjectProperty
Range:	/Core/DictionaryEntry

Referenced

- /Dataset/DatasetPackage

14.3 Vocabularies

14.3.1 ConfidentialityLevelType

Summary

Categories of confidentiality level.

Description

Describes the different confidentiality levels as given by the Traffic Light Protocol[125].

Metadata

`https://spdx.org/rdf/3.0.1/terms/Dataset/ConfidentialityLevelType`

Name:	ConfidentialityLevelType

Entries

amber Data points in the dataset can be shared only with specific organizations and their clients on a need to know basis.

clear Dataset may be distributed freely, without restriction.

green Dataset can be shared within a community of peers and partners.

red Data points in the dataset are highly confidential and can only be shared with named recipients.

14.3.2 DatasetAvailabilityType

Summary

Availability of dataset.

Description

Describes the possible types of availability of a dataset, indicating whether the dataset can be directly downloaded, can be assembled using a script for scraping the data, is only available after a clickthrough or a registration form.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Dataset/DatasetAvailabilityType`

Name:	DatasetAvailabilityType

[125] https://en.wikipedia.org/wiki/Traffic_Light_Protocol

Entries

clickthrough the dataset is not publicly available and can only be accessed after affirmatively accepting terms on a clickthrough webpage.

directDownload the dataset is publicly available and can be downloaded directly.

query the dataset is publicly available, but not all at once, and can only be accessed through queries which return parts of the dataset.

registration the dataset is not publicly available and an email registration is required before accessing the dataset, although without an affirmative acceptance of terms.

scrapingScript the dataset provider is not making available the underlying data and the dataset must be reassembled, typically using the provided script for scraping the data.

14.3.3 DatasetType

Summary

Enumeration of dataset types.

Description

Describes the different structures of data within a given dataset. A dataset can have multiple types of data, or even a single type of data but still match multiple types, for example sensor data could also be timeseries or labeled image data could also be considered categorical.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Dataset/DatasetType`

Name:	DatasetType

Entries

audio data is audio based, such as a collection of music from the 80s.

categorical data that is classified into a discrete number of categories, such as the eye color of a population of people.

graph data is in the form of a graph where entries are somehow related to each other through edges, such a social network of friends.

image data is a collection of images such as pictures of animals.

noAssertion data type is not known.

numeric data consists only of numeric entries.

other data is of a type not included in this list.

sensor data is recorded from a physical sensor, such as a thermometer reading or biometric device.

structured data is stored in tabular format or retrieved from a relational database.

syntactic data describes the syntax or semantics of a language or text, such as a parse tree used for natural language processing.

text data consists of unstructured text, such as a book, Wikipedia article (without images), or transcript.

timeseries data is recorded in an ordered sequence of timestamped entries, such as the price of a stock over the course of a day.

timestamp data is recorded with a timestamp for each entry, but not necessarily ordered or at specific intervals, such as when a taxi ride starts and ends.

video data is video based, such as a collection of movie clips featuring Tom Hanks.

This page intentionally left blank.

15 AI

Summary

The AI Profile is designed to provide a standardized way of documenting and sharing information about AI software packages (i.e. systems).

Description

The AI namespace defines a set of concepts and data elements related to AI system and model artifacts. These artifacts are the tangible outputs of the AI development process, such as software packages, models, and datasets.

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI`

Name:	AI

Profile conformance

For an element collection to be conformant with this profile, the following has to hold:

1. for every `/AI/AIPackage` there MUST exist exactly one `/Core/Relationship` of type `hasConcludedLicense` having that element as its `from` property and a `/SimpleLicensing/AnyLicenseInfo` as its `to` property.
2. for every `/AI/AIPackage` there MUST exist exactly one `/Core/Relationship` of type `hasDeclaredLicense` having that element as its `from` property and a `/SimpleLicensing/AnyLicenseInfo` as its `to` property.

15.1 Classes

15.1.1 AIPackage

Summary

Specifies an AI package and its associated information.

Description

Metadata information that can be added to a package to describe an AI application or trained AI model.

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/AIPackage`

Name:	AIPackage
Instantiability:	Concrete
SubclassOf:	/Software/Package

Superclasses

- /Software/Package
- /Software/SoftwareArtifact
- /Core/Artifact
- /Core/Element

Properties

Property	Type	minCount	maxCount
autonomyType	/Core/PresenceType	0	1
domain	xsd:string	0	*
energyConsumption	EnergyConsumption	0	1
hyperparameter	/Core/DictionaryEntry	0	*
informationAboutApplication	xsd:string	0	1
informationAboutTraining	xsd:string	0	1
limitation	xsd:string	0	1
metric	/Core/DictionaryEntry	0	*
metricDecisionThreshold	/Core/DictionaryEntry	0	*
modelDataPreprocessing	xsd:string	0	*
modelExplainability	xsd:string	0	*
safetyRiskAssessment	SafetyRiskAssessmentType	0	1
standardCompliance	xsd:string	0	*
typeOfModel	xsd:string	0	*
useSensitivePersonalInformation	/Core/PresenceType	0	1

External properties cardinality updates

Property	minCount	maxCount
/Core/Artifact/releaseTime	1	
/Core/Artifact/suppliedBy	1	
/Software/Package/downloadLocation	1	
/Software/Package/packageVersion	1	
/Software/SoftwareArtifact/primaryPurpose	1	

All properties (informative)

Property	Type	minCount	maxCount
additionalPurpose	SoftwarePurpose	0	*
attributionText	xsd:string	0	*
autonomyType	/Core/PresenceType	0	1
builtTime	DateTime	0	1
comment	xsd:string	0	1
contentIdentifier	ContentIdentifier	0	*
copyrightText	xsd:string	0	1
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
domain	xsd:string	0	*
downloadLocation	xsd:anyURI	1	1
energyConsumption	EnergyConsumption	0	1
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
homePage	xsd:anyURI	0	1
hyperparameter	/Core/DictionaryEntry	0	*
informationAboutApplication	xsd:string	0	1
informationAboutTraining	xsd:string	0	1
limitation	xsd:string	0	1
metric	/Core/DictionaryEntry	0	*
metricDecisionThreshold	/Core/DictionaryEntry	0	*
modelDataPreprocessing	xsd:string	0	*
modelExplainability	xsd:string	0	*
name	xsd:string	1	1
originatedBy	Agent	0	*
packageUrl	xsd:anyURI	0	1
packageVersion	xsd:string	1	1
primaryPurpose	SoftwarePurpose	1	1
releaseTime	DateTime	1	1
safetyRiskAssessment	SafetyRiskAssessmentType	0	1
sourceInfo	xsd:string	0	1
spdxId	xsd:anyURI	1	1
standardCompliance	xsd:string	0	*
standardName	xsd:string	0	*
summary	xsd:string	0	1
suppliedBy	Agent	1	1
supportLevel	SupportType	0	*
typeOfModel	xsd:string	0	*
useSensitivePersonalInformation	/Core/PresenceType	0	1
validUntilTime	DateTime	0	1
verifiedUsing	IntegrityMethod	0	*

15.1.2 EnergyConsumption

Summary

A class for describing the energy consumption incurred by an AI model in different stages of its lifecycle.

Description

A class to denote the known or estimated energy consumption of an AI model during its training, fine-tuning, and inference stages.

Example

```
{
  "type": "ai_EnergyConsumption",
  "ai_trainingEnergyConsumption": [
    {
      "type": "ai_EnergyConsumptionDescription",
      "ai_energyQuantity": "36.5",
      "ai_energyUnit": "kilowattHour"
    }
  ],
  "ai_inferenceEnergyConsumption": [
    {
      "type": "ai_EnergyConsumptionDescription",
      "ai_energyQuantity": "0.042",
      "ai_energyUnit": "kilowattHour"
    }
  ]
}
```

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/EnergyConsumption`

Name:	EnergyConsumption
Instantiability:	Concrete

Properties

Property	Type	minCount	maxCount
finetuningEnergyConsumption	EnergyConsumptionDescription	0	*
inferenceEnergyConsumption	EnergyConsumptionDescription	0	*
trainingEnergyConsumption	EnergyConsumptionDescription	0	*

All properties (informative)

Property	Type	minCount	maxCount
finetuningEnergyConsumption	EnergyConsumptionDescription	0	*
inferenceEnergyConsumption	EnergyConsumptionDescription	0	*
trainingEnergyConsumption	EnergyConsumptionDescription	0	*

15.1.3 EnergyConsumptionDescription

Summary

The class that helps note down the quantity of energy consumption and the unit used for measurement.

Description

This class is designed to store energy consumption data, including the quantity and the unit of measurement.

The `energyQuantity` property stores the amount of energy consumed, and the `energyUnit` property stores the unit used for measurement.

For example, 0.0042 kilowatt-hour of energy will have `0.042` as a value for property `energyQuantity`, and `"kilowattHour"` as a value for property `energyUnit`.

Example

```
{
  "type": "ai_EnergyConsumptionDescription",
  "ai_energyQuantity": "0.042",
  "ai_energyUnit": "kilowattHour"
}
```

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/EnergyConsumptionDescription`

Name:	EnergyConsumptionDescription
Instantiability:	Concrete

Properties

Property	Type	minCount	maxCount
energyQuantity	xsd:decimal	1	1
energyUnit	EnergyUnitType	1	1

All properties (informative)

Property	Type	minCount	maxCount
energyQuantity	xsd:decimal	1	1
energyUnit	EnergyUnitType	1	1

15.2 Properties

15.2.1 autonomyType

Summary

Indicates whether the system can perform a decision or action without human involvement or guidance.

Description

Indicates if the system is fully automated or a human is involved in any of the decisions of the AI system.

- yes: Indicates that the system is fully automated.
- no: Indicates that a human is involved in any of the decisions of the AI system.
- noAssertion: Makes no assertion about the autonomy.

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/autonomyType`

Name:	autonomyType
Nature:	ObjectProperty
Range:	/Core/PresenceType

Referenced

- /AI/AIPackage

15.2.2 domain

Summary

Captures the domain in which the AI package can be used.

Description

A free-form text that describes the domain where the AI model contained in the AI software can be expected to operate successfully.

Examples include computer vision, natural language processing, etc.

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/domain`

Name:	domain
Nature:	DataProperty
Range:	xsd:string

Referenced

- /AI/AIPackage

15.2.3 energyConsumption

Summary

Indicates the amount of energy consumption incurred by an AI model.

Description

Captures the energy consumption of an AI model, either known or estimated.

In the absence of direct measurements, an SPDX data creator may choose to estimate the energy consumption based on information about computational resources (e.g., number of floating-point operations), training time, and other relevant training details.

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/energyConsumption`

Name:	energyConsumption
Nature:	ObjectProperty
Range:	EnergyConsumption

Referenced

- /AI/AIPackage

15.2.4 energyQuantity

Summary

Represents the energy quantity.

Description

Provides the quantity information of the energy.

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/energyQuantity`

Name:	energyQuantity
Nature:	DataProperty
Range:	xsd:decimal

Referenced

- /AI/EnergyConsumptionDescription

15.2.5 energyUnit

Summary

Specifies the unit in which energy is measured.

Description

Provides the unit information of the energy.

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/energyUnit`

Name:	energyUnit
Nature:	ObjectProperty
Range:	EnergyUnitType

Referenced

- /AI/EnergyConsumptionDescription

15.2.6 finetuningEnergyConsumption

Summary

Specifies the amount of energy consumed when finetuning the AI model that is being used in the AI system.

Description

The field specifies the amount of energy consumed when finetuning the AI model that is being used in the AI system.

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/finetuningEnergyConsumption`

Name:	finetuningEnergyConsumption
Nature:	ObjectProperty
Range:	EnergyConsumptionDescription

Referenced

- /AI/EnergyConsumption

15.2.7 hyperparameter

Summary

Records a hyperparameter used to build the AI model contained in the AI package.

Description

Records a hyperparameter value.

Hyperparameters are settings defined before the training process that control the learning algorithm's behavior. They differ from model parameters, which are learned from the data during training. Developers typically set hyperparameters manually or through a process of hyperparameter tuning (also known as trial and error).

Examples of hyperparameters include learning rate, batch size, and the number of layers in a neural network.

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/hyperparameter`

Name:	hyperparameter
Nature:	ObjectProperty
Range:	/Core/DictionaryEntry

Referenced

- /AI/AIPackage

15.2.8 inferenceEnergyConsumption

Summary

Specifies the amount of energy consumed during inference time by an AI model that is being used in the AI system.

Description

The field specifies the amount of energy consumed during inference time by an AI model that is being used in the AI system.

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/inferenceEnergyConsumption`

Name:	inferenceEnergyConsumption
Nature:	ObjectProperty
Range:	EnergyConsumptionDescription

Referenced

- /AI/EnergyConsumption

15.2.9 informationAboutApplication

Summary

Provides relevant information about the AI software, not including the model description.

Description

A free-form text description of how the AI model is used within the software.

It should include any relevant information, such as pre-processing steps, third-party APIs, and other pertinent details.

It can also include:

- Functionality provided by the AI model within the software application, including: any specific tasks or decisions it is designed to perform; any pre-processing steps that are applied to the input data before it is fed into the AI model for inference, such as data cleaning, normalization, or feature extraction; and any third-party APIs or services that are used in conjunction with the AI model, such as data sources, cloud services, or other AI models.
- Description of any dependencies or requirements needed to run the AI model within the software application, including: specific hardware, software libraries, and operating systems.

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/informationAboutApplication`

Name:	informationAboutApplication
Nature:	DataProperty
Range:	xsd:string

Referenced

- /AI/AIPackage

15.2.10 informationAboutTraining

Summary

Describes relevant information about different steps of the training process.

Description

A detailed explanation of the training process, including the specific techniques, algorithms, and methods employed.

Examples include:

- training data used to train the AI model, along with any relevant details about its source, quality, and pre-processing steps;
- specific training algorithms employed, including stochastic gradient descent, backpropagation, and reinforcement learning;
- specific training techniques used to improve the performance or accuracy of the AI model, such as transfer learning, fine-tuning, or active learning; and
- any evaluation metrics used to assess the performance of the AI model during the training process, including accuracy, precision, recall, and F1 score.

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/informationAboutTraining`

Name:	informationAboutTraining
Nature:	DataProperty
Range:	xsd:string

Referenced

- /AI/AIPackage

15.2.11 limitation

Summary

Captures a limitation of the AI software.

Description

A free-form text that captures a limitation of the AI package (or of the AI models present in the AI package).

Note that this is not guaranteed to be exhaustive.

For instance, a limitation might be that the AI package cannot be used on datasets from a certain demography.

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/limitation`

Name:	limitation
Nature:	DataProperty
Range:	xsd:string

Referenced

- /AI/AIPackage

15.2.12 metric

Summary

Records the measurement of prediction quality of the AI model.

Description

Records the measurement with which the AI model was evaluated.

This makes statements about the prediction quality including uncertainty, accuracy, characteristics of the tested population, quality, fairness, explainability, robustness etc.

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/metric`

Name:	metric
Nature:	ObjectProperty
Range:	/Core/DictionaryEntry

Referenced

- /AI/AIPackage

15.2.13 metricDecisionThreshold

Summary

Captures the threshold that was used for computation of a metric described in the metric field.

Description

Each metric might be computed based on a decision threshold.

For instance, precision or recall is typically computed by checking if the probability of the outcome is larger than 0.5.

Each decision threshold should match with a metric field defined in the AI package.

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/metricDecisionThreshold`

Name:	metricDecisionThreshold
Nature:	ObjectProperty
Range:	/Core/DictionaryEntry

Referenced

- /AI/AIPackage

15.2.14 modelDataPreprocessing

Summary

Describes all the preprocessing steps applied to the training data before the model training.

Description

A free-form text that describes the preprocessing steps applied to the training data before training of the model(s) contained in the AI software.

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/modelDataPreprocessing`

Name:	modelDataPreprocessing
Nature:	DataProperty
Range:	xsd:string

Referenced

- /AI/AIPackage

15.2.15 modelExplainability

Summary

Describes methods that can be used to explain the results from the AI model.

Description

A free-form text that lists the different explainability mechanisms and how they can be used to explain the results from the AI model.

The mechanisms can be model-agnostic methods, such as SHapley Additive exPlanations (SHAP)[126] and Local Interpretable Model-agnostic Explanations (LIME)[127], and model-specific methods that applied to a limited category of models.

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/modelExplainability`

Name:	modelExplainability
Nature:	DataProperty
Range:	xsd:string

Referenced

- /AI/AIPackage

15.2.16 safetyRiskAssessment

Summary

Records the results of general safety risk assessment of the AI system.

Description

Records the results of general safety risk assessment of the AI system.

Using categorization according to the EU general risk assessment methodology[128]. The methodology implements Article 20 of Regulation (EC) No 765/2008 and is intended to assist authorities when they assess general product safety compliance.

It is important to note that this categorization differs from the one proposed in the EU AI Act’s provisional agreement.

[126] https://shap.readthedocs.io/

[127] https://github.com/marcotcr/lime

[128] https://ec.europa.eu/docsroom/documents/17107

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/safetyRiskAssessment`

Name:	safetyRiskAssessment
Nature:	ObjectProperty
Range:	SafetyRiskAssessmentType

Referenced

- /AI/AIPackage

15.2.17 standardCompliance

Summary

Captures a standard that is being complied with.

Description

A free-form text that captures a standard that the AI software complies with.

This includes both published and unpublished standards, such as those developed by ISO, IEEE, and ETSI.

The standard may, but is not necessarily required to, satisfy a legal or regulatory requirement.

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/standardCompliance`

Name:	standardCompliance
Nature:	DataProperty
Range:	xsd:string

Referenced

- /AI/AIPackage

15.2.18 trainingEnergyConsumption

Summary

Specifies the amount of energy consumed when training the AI model that is being used in the AI system.

Description

The field specifies the amount of energy consumed when training the AI model that is being used in the AI system.

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/trainingEnergyConsumption`

Name:	trainingEnergyConsumption
Nature:	ObjectProperty
Range:	EnergyConsumptionDescription

Referenced

- /AI/EnergyConsumption

15.2.19 typeOfModel

Summary

Records the type of the model used in the AI software.

Description

A free-form text that records the type of the AI model(s) used in the software.

For instance, if it is a supervised model, unsupervised model, reinforcement learning model or a combination of those.

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/typeOfModel`

Name:	typeOfModel
Nature:	DataProperty
Range:	xsd:string

Referenced

- /AI/AIPackage

15.2.20 useSensitivePersonalInformation

Summary

Records if sensitive personal information is used during model training or could be used during the inference.

Description

Notes if sensitive personal information is used in the training or inference of the AI models.

This might include biometric data, addresses or other data that can be used to infer a person's identity.

Related: `hasSensitivePersonalInformation` in `/Dataset/DatasetPackage`

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/useSensitivePersonalInformation`

Name:	useSensitivePersonalInformation
Nature:	ObjectProperty
Range:	/Core/PresenceType

Referenced

- /AI/AIPackage

15.3 Vocabularies

15.3.1 EnergyUnitType

Summary

Specifies the unit of energy consumption.

Description

List the different acceptable units for measuring energy consumption.

If the unit in which the energy consumption has been recorded is not listed here, please select "other".

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/EnergyUnitType`

Name:	EnergyUnitType

Entries

kilowattHour Kilowatt-hour.

megajoule Megajoule.

other Any other units of energy measurement.

15.3.2 SafetyRiskAssessmentType

Summary

Specifies the safety risk level.

Description

Lists the different general safety risk levels that can be used to describe the general safety risk of an AI system.

Using categorization according to the EU general risk assessment methodology[129]. The methodology implements Article 20 of Regulation (EC) No 765/2008 and is intended to assist authorities when they assess general product safety compliance.

Metadata

`https://spdx.org/rdf/3.0.1/terms/AI/SafetyRiskAssessmentType`

Name:	SafetyRiskAssessmentType

Entries

high The second-highest level of risk posed by an AI system.

low Low/no risk is posed by an AI system.

medium The third-highest level of risk posed by an AI system.

serious The highest level of risk posed by an AI system.

[129] https://ec.europa.eu/docsroom/documents/17107

This page intentionally left blank.

16 Build

Summary

The Build Profile defines the set of information required to describe an instance of a Software Build.

Description

A Software Build is defined here as the act of converting software inputs into software artifacts using software build tools. Inputs can include source code, config files, artifacts that are build environments, and build tools. Outputs can include intermediate artifacts to other build inputs or the final artifacts.

The Build profile provides a subclass of Element called Build.

It also provides a minimum set of required Relationship Types from the Core profile:

- hasInput: Describes the relationship from the Build element to its inputs.
- hasOutput: Describes the relationship from the Build element to its outputs.
- invokedBy: Describes the relationship from the Build element to the Agent that invoked it.

In addition, the following Relationship Types may be used to describe a Build.

- hasHost: Describes the relationship from the Build element to the build stage or host.
- configures: Describes the relationship from a configuration to the Build element.
- ancestorOf: Describes a relationship from a Build element to Build elements that describe its child builds.
- descendantOf: Describes a relationship from a child Build element to its parent.
- usesTool: Describes a relationship from a Build element to a build tool.

All relationships in the Build Profile are scoped to the "build" LifecycleScopeType period.

The `hasInput` relationship can be applied to a config file or a build tool if the nature of these inputs are not known at the creation of an SPDX document.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Build`

Name:	Build

16.1 Classes

16.1.1 Build

Summary

Class that describes a build instance of software/artifacts.

Description

A build is a representation of the process in which a piece of software or artifact is built. It encapsulates information related to a build process and provides an element from which relationships can be created to describe the build's inputs, outputs, and related entities (e.g. builders, identities, etc.).

ExternalIdentifier of type "urlScheme" may be used to identify build logs. In this case, the comment of the ExternalIdentifier should be "LogReference".

Note that buildStartTime and buildEndTime are optional, and may be omitted to simplify creating reproducible builds.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Build/Build`

Name:	Build
Instantiability:	Concrete
SubclassOf:	/Core/Element

Superclasses

- /Core/Element

Properties

Property	Type	minCount	maxCount
buildEndTime	/Core/DateTime	0	1
buildId	xsd:string	0	1
buildStartTime	/Core/DateTime	0	1
buildType	xsd:anyURI	1	1
configSourceDigest	/Core/Hash	0	*
configSourceEntrypoint	xsd:string	0	*
configSourceUri	xsd:anyURI	0	*
environment	/Core/DictionaryEntry	0	*
parameter	/Core/DictionaryEntry	0	*

All properties (informative)

Property	Type	minCount	maxCount
buildEndTime	/Core/DateTime	0	1
buildId	xsd:string	0	1
buildStartTime	/Core/DateTime	0	1
buildType	xsd:anyURI	1	1
comment	xsd:string	0	1
configSourceDigest	/Core/Hash	0	*
configSourceEntrypoint	xsd:string	0	*
configSourceUri	xsd:anyURI	0	*
creationInfo	CreationInfo	1	1
description	xsd:string	0	1
environment	/Core/DictionaryEntry	0	*
extension	/Extension/Extension	0	*
externalIdentifier	ExternalIdentifier	0	*
externalRef	ExternalRef	0	*
name	xsd:string	0	1
parameter	/Core/DictionaryEntry	0	*
spdxId	xsd:anyURI	1	1
summary	xsd:string	0	1
verifiedUsing	IntegrityMethod	0	*

16.2 Properties

16.2.1 buildEndTime

Summary

Property that describes the time at which a build stops.

Description

buildEndTime describes the time at which a build stops or finishes.

This value is typically recorded by the builder.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Build/buildEndTime`

Name:	buildEndTime
Nature:	DataProperty
Range:	/Core/DateTime

Referenced

- /Build/Build

16.2.2 buildId

Summary

A buildId is a locally unique identifier used by a builder to identify a unique instance of a build produced by it.

Description

A buildId is a locally unique identifier to identify a unique instance of a build, according to the buildType.

This identifier differs based on build toolchain, platform, or naming convention used by an organization or standard.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Build/buildId`

Name:	buildId
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Build/Build

16.2.3 buildStartTime

Summary

Property describing the start time of a build.

Description

buildStartTime is the time at which a build is triggered.

The builder typically records this value.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Build/buildStartTime`

Name:	buildStartTime
Nature:	DataProperty
Range:	/Core/DateTime

Referenced

- /Build/Build

16.2.4 buildType

Summary

A buildType is a hint that is used to indicate the toolchain, platform, or infrastructure that the build was invoked on.

Description

A buildType is an IRI expressing the toolchain, platform, or infrastructure that the build was invoked on.

The buildType is used to interpret the meaning of other build parameters by defining the “type” of build; if the same buildType is seen in different Build elements, it means they are the same kind of build, but difference instances and possible with different configurations.

If you are not using a well-known buildType, it should be namespaced to a domain you own to prevent conflicts with other buildType IRIs.

Examples of a buildType might be:

- A GitHub action workflow
- A step in a GitHub actions pipeline

- An invocation of a compiler or other tool
- A script that orchestrates builds at a higher level

Keep in mind that builds can be “nested” using the `ancestorOf` relationship.

If the buildType IRI is not recognized, it is still possible to inspect other properties of the build, but it may not be possible to derive deeper meaning from them.

For more information, see the SLSA definition of buildType.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Build/buildType`

Name:	buildType
Nature:	DataProperty
Range:	xsd:anyURI

Referenced

- /Build/Build

16.2.5 configSourceDigest

Summary

Property that describes the digest of the build configuration file used to invoke a build.

Description

configSourceDigest is the checksum of the build configuration file used by a builder to execute a build, according to the buildType.

This Property uses the Core model’s Hash[130] class.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Build/configSourceDigest`

Name:	configSourceDigest
Nature:	ObjectProperty
Range:	/Core/Hash

Referenced

- /Build/Build

16.2.6 configSourceEntrypoint

Summary

Property describes the invocation entrypoint of a build.

Description

A build entrypoint is the invoked executable of a build which always runs when the build is triggered, according to the buildType.

For example, when a build is triggered by running a shell script, the entrypoint is `script.sh`.

In terms of a declared build, the entrypoint is the position in a configuration file or a build declaration which is always run when the build is triggered.

For example, in the following configuration file, the entrypoint of the build is `publish`.

```
name: Publish packages to PyPI

on:
  create:
```

[130] ../../Core/Classes/Hash.md

```
    tags: "*"

jobs:
  publish:
    runs-on: ubuntu-latest
    if: startsWith(github.ref, 'refs/tags/')
    steps:

...
```

Metadata

`https://spdx.org/rdf/3.0.1/terms/Build/configSourceEntrypoint`

Name:	configSourceEntrypoint
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Build/Build

16.2.7 configSourceUri

Summary

Property that describes the URI of the build configuration source file.

Description

If a build configuration exists for the toolchain or platform performing the build, the configSourceUri of a build is the URI of that build configuration, according to the buildType.

For example, a build triggered by a GitHub Action is defined by a build configuration YAML file. In this case, the configSourceUri is the URL of that YAML file.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Build/configSourceUri`

Name:	configSourceUri
Nature:	DataProperty
Range:	xsd:anyURI

Referenced

- /Build/Build

16.2.8 environment

Summary

Property describing the session in which a build is invoked.

Description

environment is a map of environment variables and values that are set during a build session, according to the buildType.

This is different from the parameter[131] property in that it describes the environment variables set before a build is invoked rather than the variables provided to the builder.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Build/environment`

Name:	environment
Nature:	ObjectProperty
Range:	/Core/DictionaryEntry

[131] parameter.md

Referenced

- /Build/Build

16.2.9 parameter

Summary

Property describing a parameter used in an instance of a build.

Description

parameter is a key-value of a build parameter and its value that was provided to the builder for a build instance, according to the buildType.

This is different from the environment[132] property in that the key and value are provided as command line arguments or a configuration file to the builder.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Build/parameter`

Name:	parameter
Nature:	ObjectProperty
Range:	/Core/DictionaryEntry

Referenced

- /Build/Build

[132] environment.md

17 Lite

Summary

The SPDX Lite profile defines a simple view of SPDX data, from the point of view of use cases in some industries.

Description

The SPDX Lite profile consists of mandatory and recommended information.

The mandatory data in SPDX Lite is basic but useful for complying with licenses. It is easy to understand licensing information by reading an SPDX Lite file.

SPDX Lite aims at a balance between the full SPDX data model and actual workflows in some industries.

An SPDX Lite document can also be used in parallel with other SPDX documents in software supply chains.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Lite`

Name:	Lite

Profile conformance

In addition to the following mandatory requirements, please refer to the corresponding Annex for elements that should be included as part of a document conforming to the Lite profile.

For a `/Software/Package` to be conformant with this profile, the following has to hold:

1. The minCount for `copyrightText` is 1
2. The minCount for `packageVersion` is 1
3. The minCount for `suppliedBy` is 1
4. At least one of `downloadLocation` or `packageUrl` must be present

Additionally:

1. for every `/Software/Package` there MUST exist exactly one `/Core/Relationship` of type `hasConcludedLicense` having that element as its `from` property and a `/SimpleLicensing/AnyLicenseInfo` as its `to` property.
2. for every `/Software/Package` there MUST exist exactly one `/Core/Relationship` of type `hasDeclaredLicense` having that element as its `from` property and a `/SimpleLicensing/AnyLicenseInfo` as its `to` property.

For a `/Core/SpdxDocument` to be conformant with this profile, the following has to hold:

1. The minCount for `element` is 1
2. The minCount for `rootElement` is 1

For a `/Software/Sbom` to be conformant with this profile, the following has to hold:

1. The minCount for `element` is 1
2. The minCount for `rootElement` is 1

Finally, for a `/Core/Agent` to be conformant with this profile, the following has to hold:

1. The minCount for `name` is 1

This page intentionally left blank.

18 Extension

Summary

Everything having to do with SPDX extensions.

Description

The Extension namespace defines the abstract Extension class serving as the base for all defined extension subclasses.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Extension`

Name:	Extension

18.1 Classes

18.1.1 CdxPropertiesExtension

Summary

A type of extension consisting of a list of name value pairs.

Description

This extension provides a more structured extension using a name-value approach.

Unlike key-value stores, properties in CdxPropertiesExtension support duplicate names, each potentially having different values.

This is intended to be compatible with the CycloneDX property `properties`.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Extension/CdxPropertiesExtension`

Name:	CdxPropertiesExtension
Instantiability:	Concrete
SubclassOf:	Extension

Superclasses

- /Extension/Extension

Properties

Property	Type	minCount	maxCount
cdxProperty	CdxPropertyEntry	1	*

All properties (informative)

Property	Type	minCount	maxCount
cdxProperty	CdxPropertyEntry	1	*

18.1.2 CdxPropertyEntry

Summary

A property name with an associated value.

Description

Each CdxPropertyEntry contains a name-value pair which maps the name to its associated value.

Unlike key-value stores, properties in CdxPropertiesExtension support duplicate names, each potentially having different values.

This class can be used to implement CycloneDX compatible properties.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Extension/CdxPropertyEntry`

Name:	CdxPropertyEntry
Instantiability:	Concrete

Properties

Property	Type	minCount	maxCount
cdxPropName	xsd:string	1	1
cdxPropValue	xsd:string	0	1

All properties (informative)

Property	Type	minCount	maxCount
cdxPropName	xsd:string	1	1
cdxPropValue	xsd:string	0	1

18.1.3 Extension

Summary

A characterization of some aspect of an Element that is associated with the Element in a generalized fashion.

Description

An Extension is a characterization of some aspect of an Element that is associated with the Element in a generalized fashion.

Rather than being associated with a particular Element through the typical use of a purpose-specific object property an Extension is associated with the Element it characterizes using a single common generalized object property.

This approach serves multiple purposes:

1. Support profile-based extended characterization of Elements. Enables specification and expression of Element characterization extensions within any profile and namespace of SPDX without requiring changes to other profiles or namespaces and without requiring local subclassing of remote classes (which could inhibit ecosystem interoperability in some cases).

2. Support extension of SPDX by adopting individuals or communities with Element characterization details uniquely specialized to their particular context. Enables adopting individuals or communities to utilize SPDX expressive capabilities along with expressing more arcane Element characterization details specific to them and not appropriate for standardization across SPDX.

3. Support structured capture of expressive solutions for gaps in SPDX coverage from real-world use. Enables adopting individuals or communities to express Element characterization details they require that are not currently defined in SPDX but likely should be. Enables a practical pipeline that:

   - identifies gaps in SPDX that should be filled,
   - expresses solutions to those gaps in a way that allows the identifying adopters to use the extended solutions with SPDX and does not conflict with current SPDX,
   - can be clearly detected among the SPDX content exchange ecosystem,
   - provides a clear and structured definition of gap solution that can be used as submission for revision to SPDX standard

Metadata

`https://spdx.org/rdf/3.0.1/terms/Extension/Extension`

Name:	Extension
Instantiability:	Abstract

18.2 Properties

18.2.1 cdxPropName

Summary

A name used in a CdxPropertyEntry name-value pair.

Description

A cdxPropName is used in a CdxPropertyEntry name-value pair.

Unlike key-value stores, properties in CdxPropertiesExtension support duplicate names, each potentially having different values.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Extension/cdxPropName`

Name:	cdxPropName
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Extension/CdxPropertyEntry

18.2.2 cdxPropValue

Summary

A value used in a CdxPropertyEntry name-value pair.

Description

A cdxPropValue is used in a CdxPropertyEntry name-value pair.

Unlike key-value stores, properties in CdxPropertiesExtension support duplicate names, each potentially having different values.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Extension/cdxPropValue`

Name:	cdxPropValue
Nature:	DataProperty
Range:	xsd:string

Referenced

- /Extension/CdxPropertyEntry

18.2.3 cdxProperty

Summary

Provides a map of a property names to a values.

Description

This field provides a mapping of a name to a value.

This is intended to be compatible with the CycloneDX property `properties`.

Unlike key-value stores, properties in CdxPropertiesExtension support duplicate names, each potentially having different values.

Metadata

`https://spdx.org/rdf/3.0.1/terms/Extension/cdxProperty`

Name:	cdxProperty
Nature:	ObjectProperty
Range:	CdxPropertyEntry

Referenced

- /Extension/CdxPropertiesExtension

Annex A

RDF model definition and diagrams (Informative)

A.1 Model definition

The SPDX RDF ontology is expressed in RDF/OWL/SHACL format and is published in online at SPDX 3.0.1 Model[1]

A.2 Diagrams

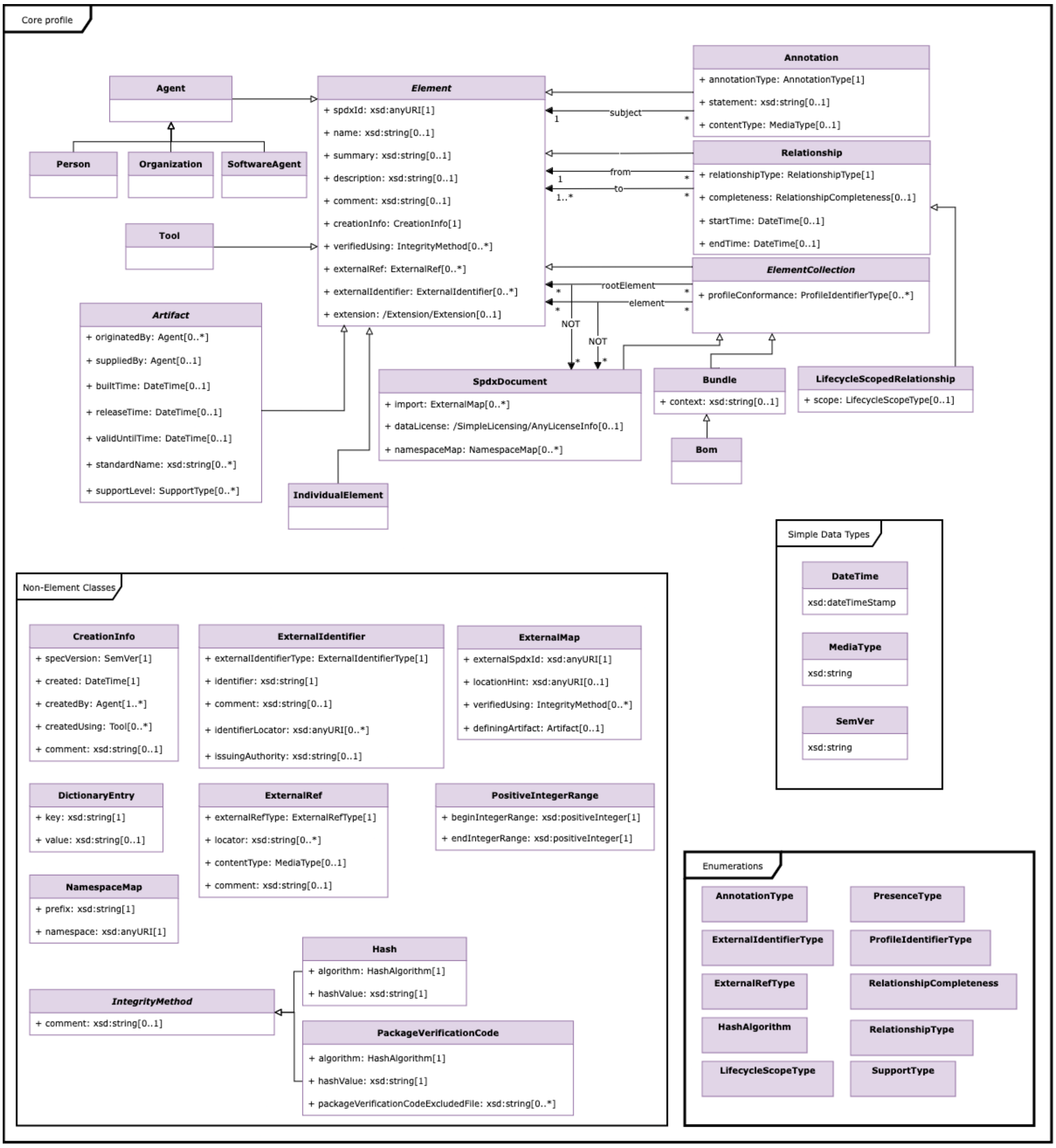
Core profile
Agent
Person
Organization
SoftwareAgent
Tool
Element
+ spdxId: xsd:anyURI[1]
+ name: xsd:string[0..1]
+ summary: xsd:string[0..1]
+ description: xsd:string[0..1]
+ comment: xsd:string[0..1]
+ creationInfo: CreationInfo[1]
+ verifiedUsing: IntegrityMethod[0..*]
+ externalRef: ExternalRef[0..*]
+ externalIdentifier: ExternalIdentifier[0..*]
+ extension: /Extension/Extension[0..1]
Annotation
+ annotationType: AnnotationType[1]
+ statement: xsd:string[0..1]
+ contentType: MediaType[0..1]
subject
1
*
Relationship
+ relationshipType: RelationshipType[1]
+ completeness: RelationshipCompleteness[0..1]
+ startTime: DateTime[0..1]
+ endTime: DateTime[0..1]
from
to
1
1..*
*
*
ElementCollection
+ profileConformance: ProfileIdentifierType[0..*]
rootElement
element
*
*
*
*
NOT
NOT
*
*
Artifact
+ originatedBy: Agent[0..*]
+ suppliedBy: Agent[0..1]
+ builtTime: DateTime[0..1]
+ releaseTime: DateTime[0..1]
+ validUntilTime: DateTime[0..1]
+ standardName: xsd:string[0..*]
+ supportLevel: SupportType[0..*]
SpdxDocument
+ import: ExternalMap[0..*]
+ dataLicense: /SimpleLicensing/AnyLicenseInfo[0..1]
+ namespaceMap: NamespaceMap[0..*]
Bundle
+ context: xsd:string[0..1]
Bom
LifecycleScopedRelationship
+ scope: LifecycleScopeType[0..1]
IndividualElement
Simple Data Types
DateTime
xsd:dateTimeStamp
MediaType
xsd:string
SemVer
xsd:string
Non-Element Classes
CreationInfo
+ specVersion: SemVer[1]
+ created: DateTime[1]
+ createdBy: Agent[1..*]
+ createdUsing: Tool[0..*]
+ comment: xsd:string[0..1]
ExternalIdentifier
+ externalIdentifierType: ExternalIdentifierType[1]
+ identifier: xsd:string[1]
+ comment: xsd:string[0..1]
+ identifierLocator: xsd:anyURI[0..*]
+ issuingAuthority: xsd:string[0..1]
ExternalMap
+ externalSpdxId: xsd:anyURI[1]
+ locationHint: xsd:anyURI[0..1]
+ verifiedUsing: IntegrityMethod[0..*]
+ definingArtifact: Artifact[0..1]
DictionaryEntry
+ key: xsd:string[1]
+ value: xsd:string[0..1]
ExternalRef
+ externalRefType: ExternalRefType[1]
+ locator: xsd:string[0..*]
+ contentType: MediaType[0..1]
+ comment: xsd:string[0..1]
PositiveIntegerRange
+ beginIntegerRange: xsd:positiveInteger[1]
+ endIntegerRange: xsd:positiveInteger[1]
NamespaceMap
+ prefix: xsd:string[1]
+ namespace: xsd:anyURI[1]
Hash
+ algorithm: HashAlgorithm[1]
+ hashValue: xsd:string[1]
IntegrityMethod
+ comment: xsd:string[0..1]
PackageVerificationCode
+ algorithm: HashAlgorithm[1]
+ hashValue: xsd:string[1]
+ packageVerificationCodeExcludedFile: xsd:string[0..*]
Enumerations
AnnotationType
PresenceType
ExternalIdentifierType
ProfileIdentifierType
ExternalRefType
RelationshipCompleteness
HashAlgorithm
RelationshipType
LifecycleScopeType
SupportType

Figure A.1: Core Profile

[1] https://spdx.org/rdf/3.0.1/spdx-model.ttl

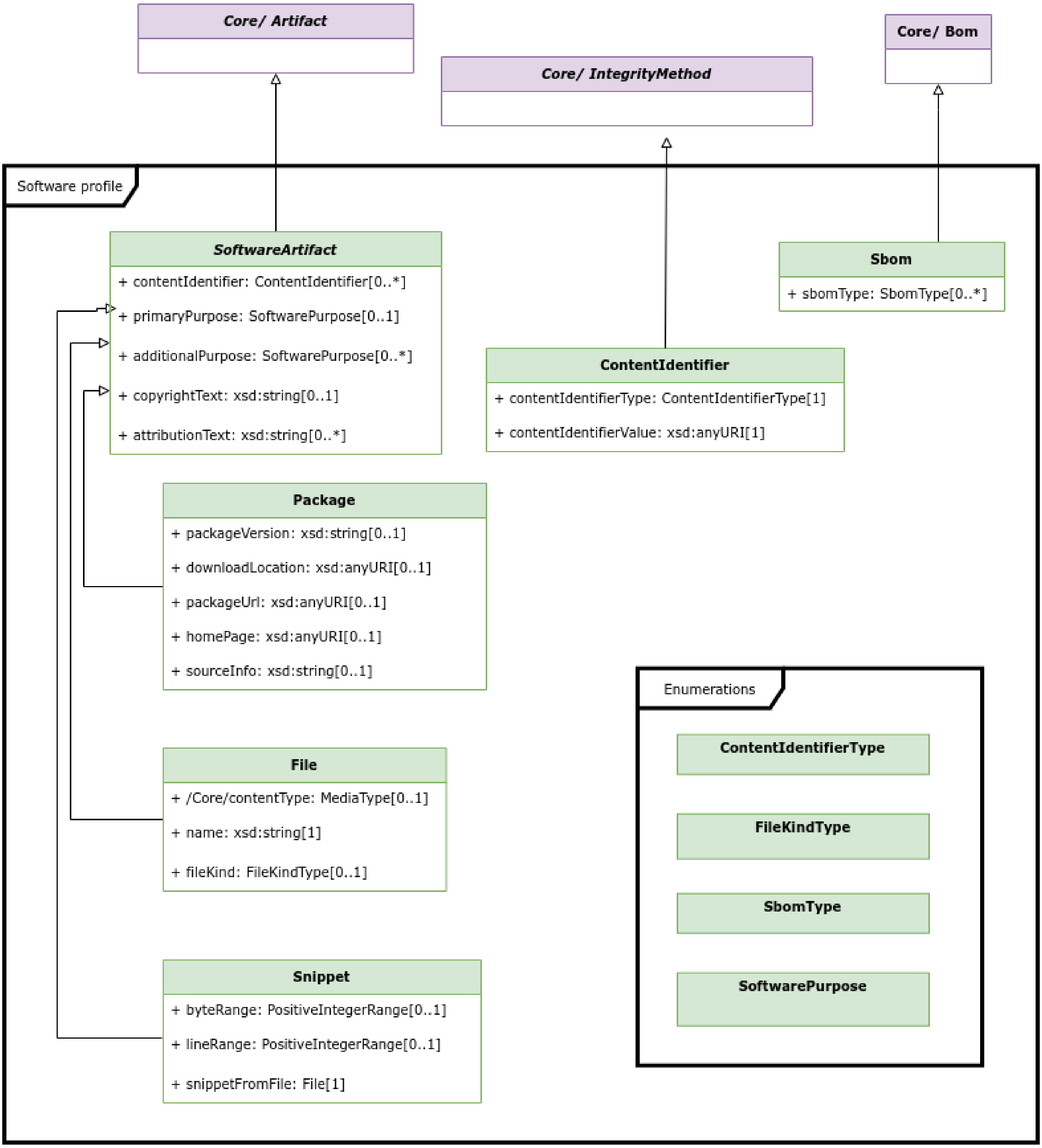
Core/ Artifact
Core/ IntegrityMethod
Core/ Bom
Software profile
SoftwareArtifact
+ contentIdentifier: ContentIdentifier[0..*]
+ primaryPurpose: SoftwarePurpose[0..1]
+ additionalPurpose: SoftwarePurpose[0..*]
+ copyrightText: xsd:string[0..1]
+ attributionText: xsd:string[0..*]
Sbom
+ sbomType: SbomType[0..*]
ContentIdentifier
+ contentIdentifierType: ContentIdentifierType[1]
+ contentIdentifierValue: xsd:anyURI[1]
Package
+ packageVersion: xsd:string[0..1]
+ downloadLocation: xsd:anyURI[0..1]
+ packageUrl: xsd:anyURI[0..1]
+ homePage: xsd:anyURI[0..1]
+ sourceInfo: xsd:string[0..1]
File
+ /Core/contentType: MediaType[0..1]
+ name: xsd:string[1]
+ fileKind: FileKindType[0..1]
Snippet
+ byteRange: PositiveIntegerRange[0..1]
+ lineRange: PositiveIntegerRange[0..1]
+ snippetFromFile: File[1]
Enumerations
ContentIdentifierType
FileKindType
SbomType
SoftwarePurpose

Figure A.2: Software Profile

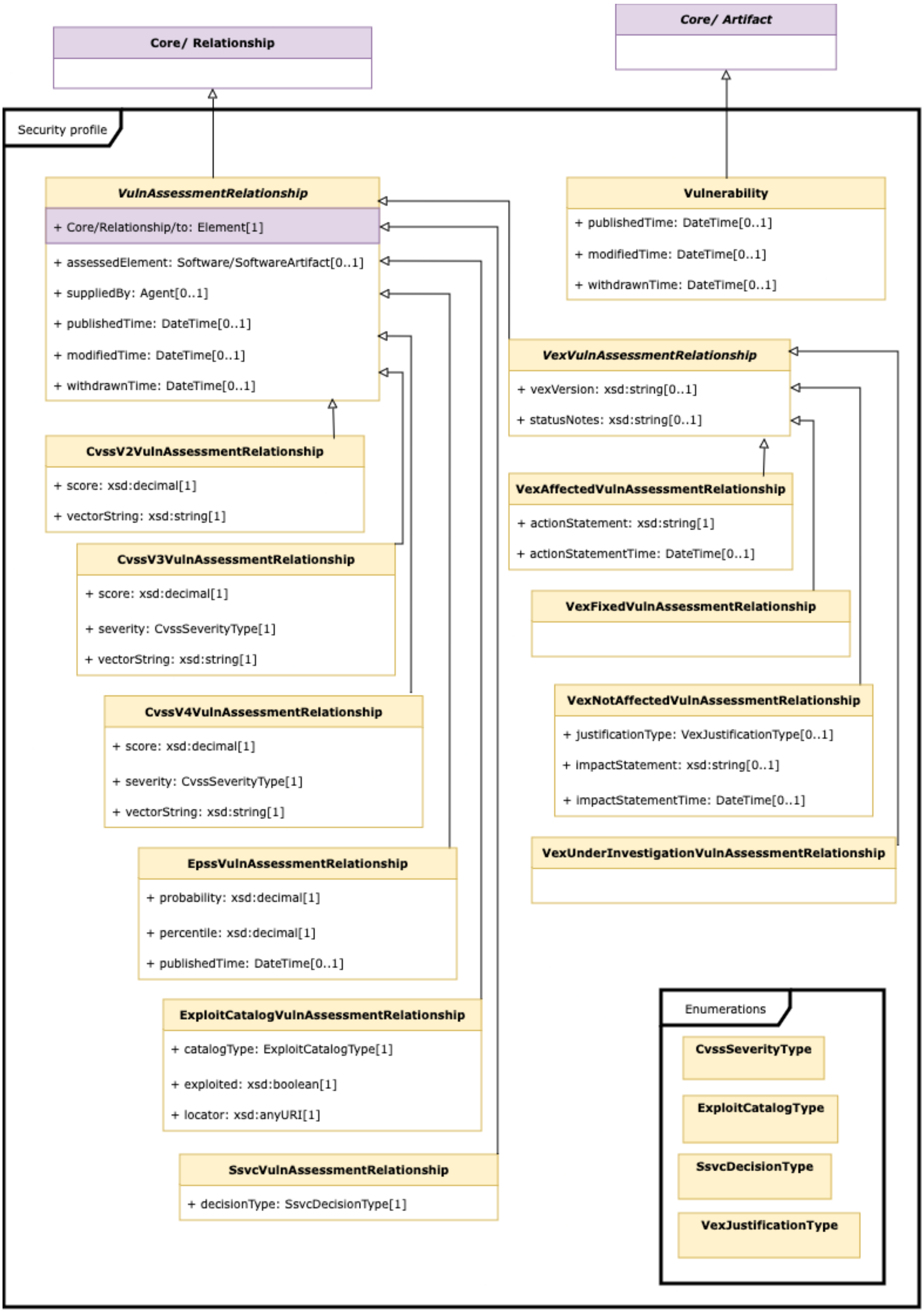
Core/ Relationship
Core/ Artifact
Security profile
VulnAssessmentRelationship
+ Core/Relationship/to: Element[1]
+ assessedElement: Software/SoftwareArtifact[0..1]
+ suppliedBy: Agent[0..1]
+ publishedTime: DateTime[0..1]
+ modifiedTime: DateTime[0..1]
+ withdrawnTime: DateTime[0..1]
Vulnerability
+ publishedTime: DateTime[0..1]
+ modifiedTime: DateTime[0..1]
+ withdrawnTime: DateTime[0..1]
VexVulnAssessmentRelationship
+ vexVersion: xsd:string[0..1]
+ statusNotes: xsd:string[0..1]
CvssV2VulnAssessmentRelationship
+ score: xsd:decimal[1]
+ vectorString: xsd:string[1]
VexAffectedVulnAssessmentRelationship
+ actionStatement: xsd:string[1]
+ actionStatementTime: DateTime[0..1]
CvssV3VulnAssessmentRelationship
+ score: xsd:decimal[1]
+ severity: CvssSeverityType[1]
+ vectorString: xsd:string[1]
VexFixedVulnAssessmentRelationship
CvssV4VulnAssessmentRelationship
+ score: xsd:decimal[1]
+ severity: CvssSeverityType[1]
+ vectorString: xsd:string[1]
VexNotAffectedVulnAssessmentRelationship
+ justificationType: VexJustificationType[0..1]
+ impactStatement: xsd:string[0..1]
+ impactStatementTime: DateTime[0..1]
VexUnderInvestigationVulnAssessmentRelationship
EpssVulnAssessmentRelationship
+ probability: xsd:decimal[1]
+ percentile: xsd:decimal[1]
+ publishedTime: DateTime[0..1]
ExploitCatalogVulnAssessmentRelationship
+ catalogType: ExploitCatalogType[1]
+ exploited: xsd:boolean[1]
+ locator: xsd:anyURI[1]
SsvcVulnAssessmentRelationship
+ decisionType: SsvcDecisionType[1]
Enumerations
CvssSeverityType
ExploitCatalogType
SsvcDecisionType
VexJustificationType

Figure A.3: Security Profile

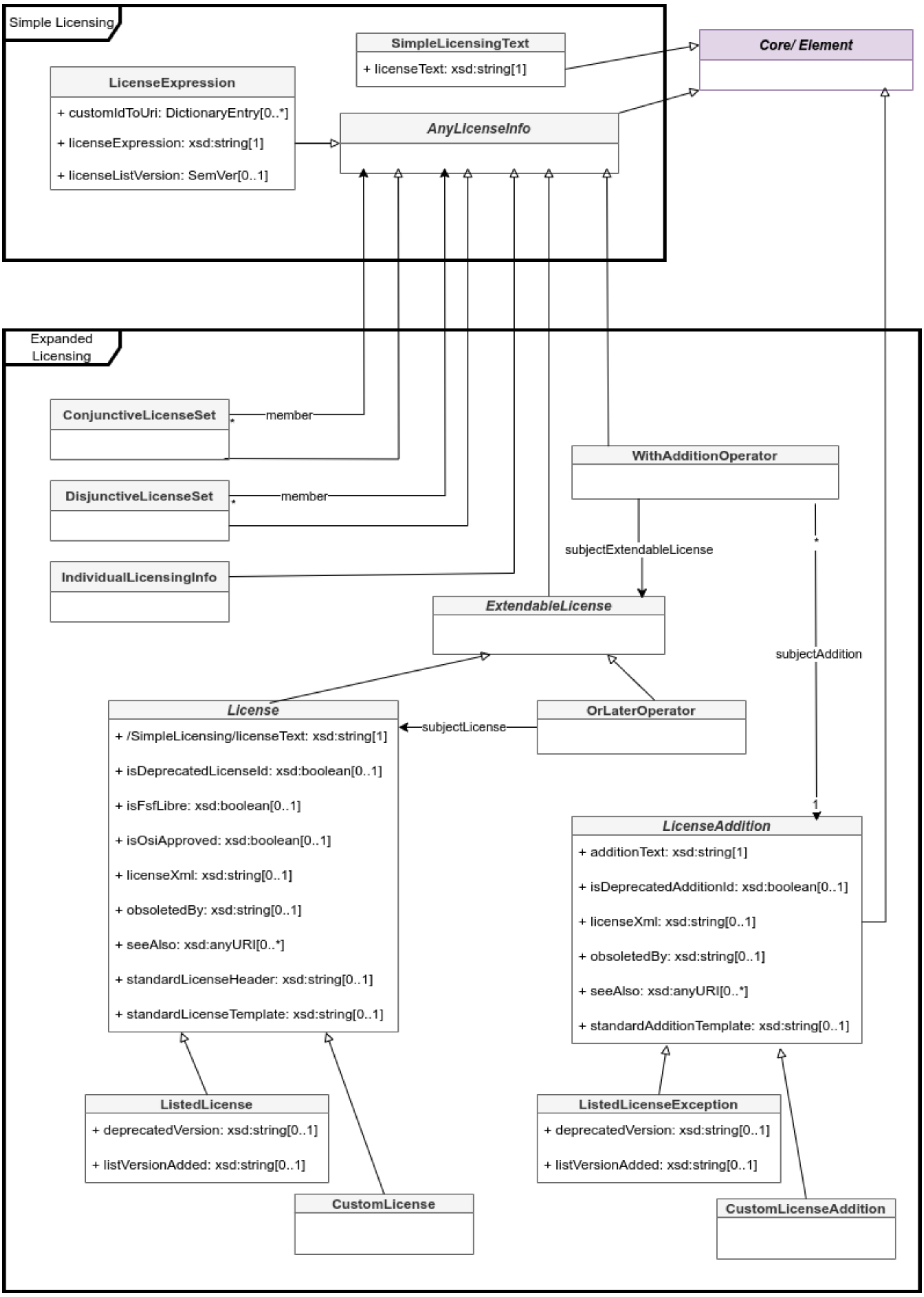
Simple Licensing
SimpleLicensingText
+ licenseText: xsd:string[1]
Core/ Element
LicenseExpression
+ customIdToUri: DictionaryEntry[0..*]
+ licenseExpression: xsd:string[1]
+ licenseListVersion: SemVer[0..1]
AnyLicenseInfo
Expanded Licensing
ConjunctiveLicenseSet
member
*
DisjunctiveLicenseSet
member
*
IndividualLicensingInfo
WithAdditionOperator
subjectExtendableLicense
*
ExtendableLicense
subjectAddition
License
+ /SimpleLicensing/licenseText: xsd:string[1]
+ isDeprecatedLicenseId: xsd:boolean[0..1]
+ isFsfLibre: xsd:boolean[0..1]
+ isOsiApproved: xsd:boolean[0..1]
+ licenseXml: xsd:string[0..1]
+ obsoletedBy: xsd:string[0..1]
+ seeAlso: xsd:anyURI[0..*]
+ standardLicenseHeader: xsd:string[0..1]
+ standardLicenseTemplate: xsd:string[0..1]
subjectLicense
OrLaterOperator
1
LicenseAddition
+ additionText: xsd:string[1]
+ isDeprecatedAdditionId: xsd:boolean[0..1]
+ licenseXml: xsd:string[0..1]
+ obsoletedBy: xsd:string[0..1]
+ seeAlso: xsd:anyURI[0..*]
+ standardAdditionTemplate: xsd:string[0..1]
ListedLicense
+ deprecatedVersion: xsd:string[0..1]
+ listVersionAdded: xsd:string[0..1]
CustomLicense
ListedLicenseException
+ deprecatedVersion: xsd:string[0..1]
+ listVersionAdded: xsd:string[0..1]
CustomLicenseAddition

Figure A.4: Licensing Profile

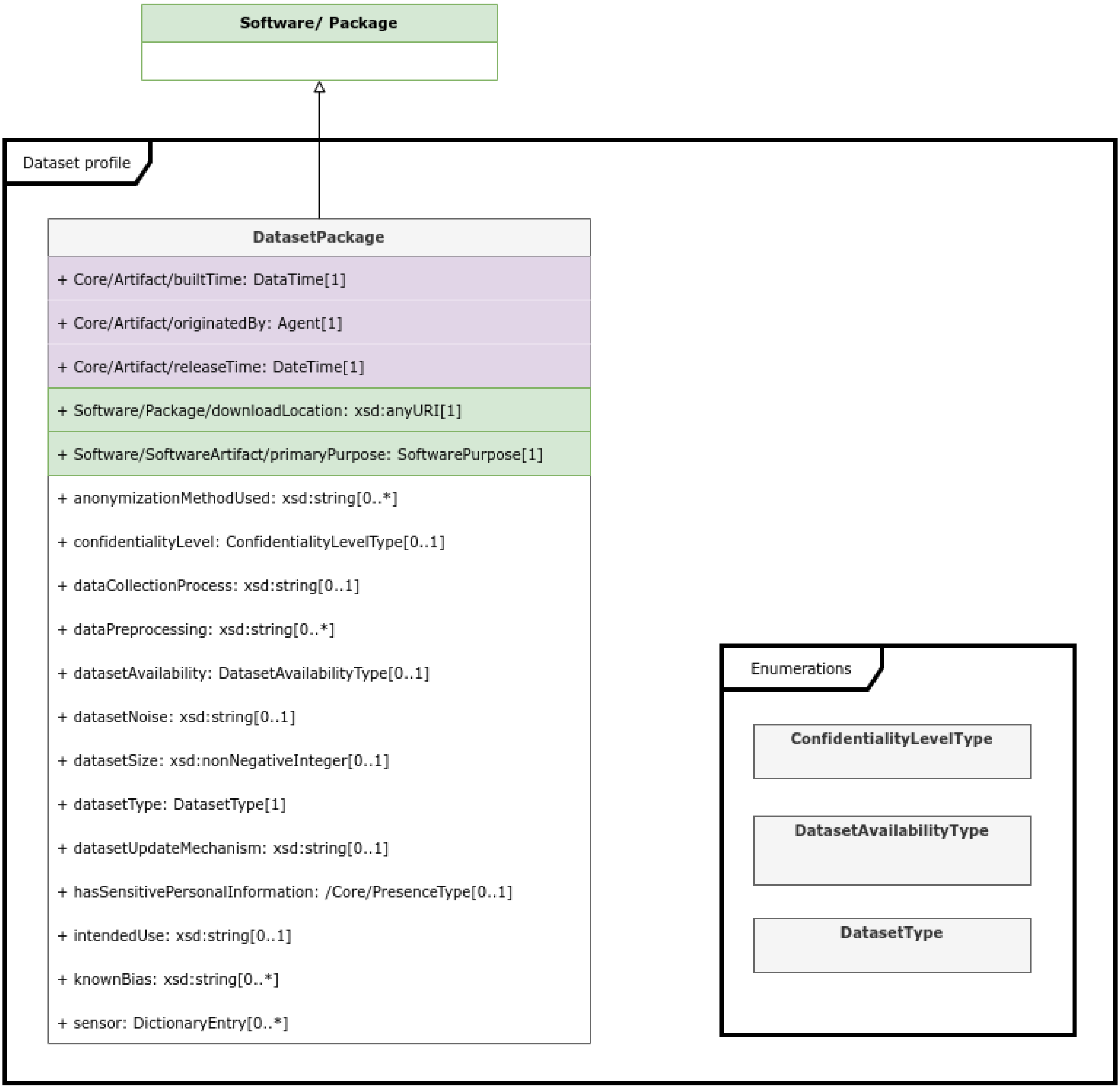
Software/ Package
Dataset profile
DatasetPackage
+ Core/Artifact/builtTime: DataTime[1]
+ Core/Artifact/originatedBy: Agent[1]
+ Core/Artifact/releaseTime: DateTime[1]
+ Software/Package/downloadLocation: xsd:anyURI[1]
+ Software/SoftwareArtifact/primaryPurpose: SoftwarePurpose[1]
+ anonymizationMethodUsed: xsd:string[0..*]
+ confidentialityLevel: ConfidentialityLevelType[0..1]
+ dataCollectionProcess: xsd:string[0..1]
+ dataPreprocessing: xsd:string[0..*]
+ datasetAvailability: DatasetAvailabilityType[0..1]
+ datasetNoise: xsd:string[0..1]
+ datasetSize: xsd:nonNegativeInteger[0..1]
+ datasetType: DatasetType[1]
+ datasetUpdateMechanism: xsd:string[0..1]
+ hasSensitivePersonalInformation: /Core/PresenceType[0..1]
+ intendedUse: xsd:string[0..1]
+ knownBias: xsd:string[0..*]
+ sensor: DictionaryEntry[0..*]
Enumerations
ConfidentialityLevelType
DatasetAvailabilityType
DatasetType

Figure A.5: Dataset Profile

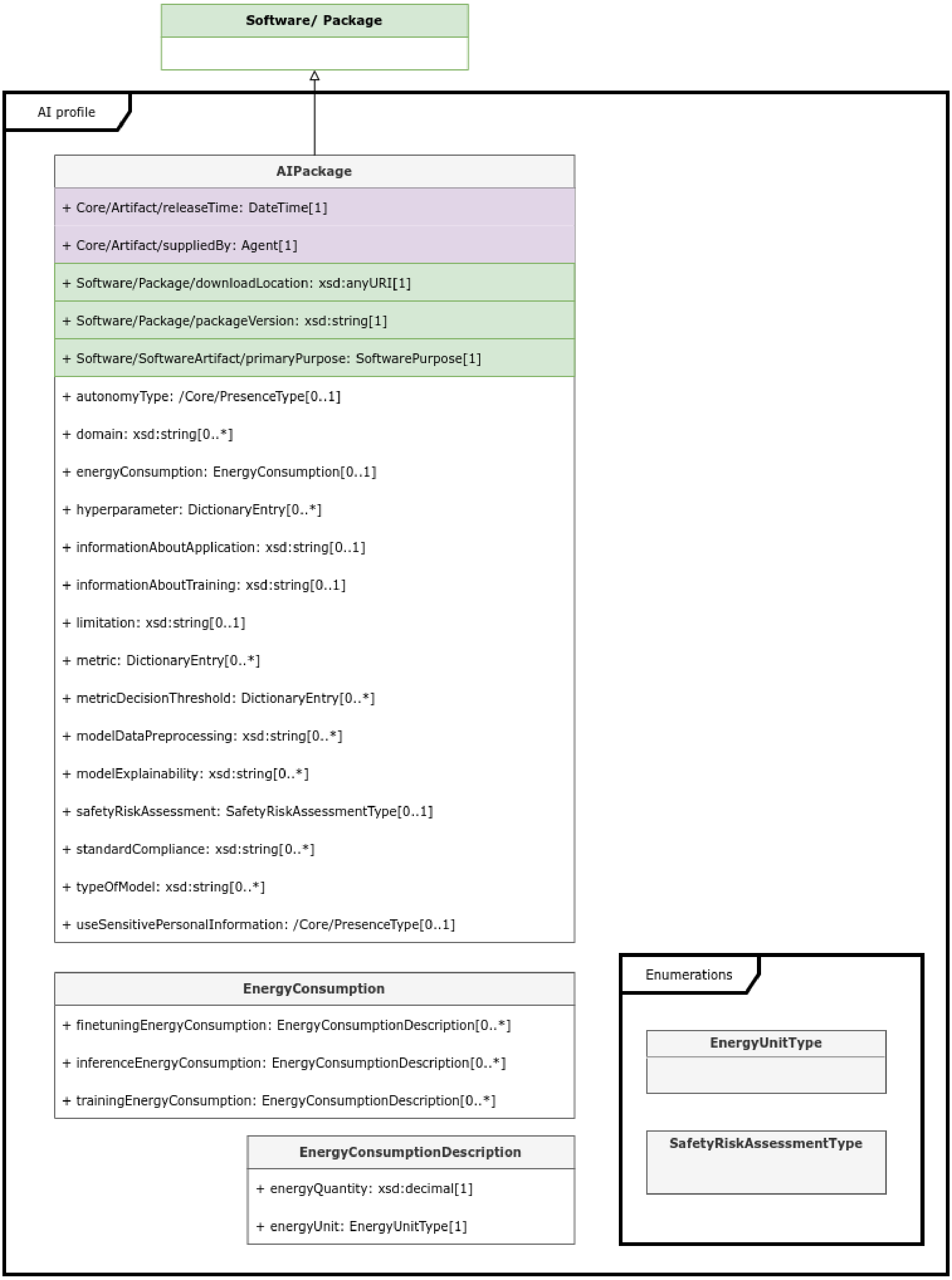
Software/ Package
AI profile
AIPackage
+ Core/Artifact/releaseTime: DateTime[1]
+ Core/Artifact/suppliedBy: Agent[1]
+ Software/Package/downloadLocation: xsd:anyURI[1]
+ Software/Package/packageVersion: xsd:string[1]
+ Software/SoftwareArtifact/primaryPurpose: SoftwarePurpose[1]
+ autonomyType: /Core/PresenceType[0..1]
+ domain: xsd:string[0..*]
+ energyConsumption: EnergyConsumption[0..1]
+ hyperparameter: DictionaryEntry[0..*]
+ informationAboutApplication: xsd:string[0..1]
+ informationAboutTraining: xsd:string[0..1]
+ limitation: xsd:string[0..1]
+ metric: DictionaryEntry[0..*]
+ metricDecisionThreshold: DictionaryEntry[0..*]
+ modelDataPreprocessing: xsd:string[0..*]
+ modelExplainability: xsd:string[0..*]
+ safetyRiskAssessment: SafetyRiskAssessmentType[0..1]
+ standardCompliance: xsd:string[0..*]
+ typeOfModel: xsd:string[0..*]
+ useSensitivePersonalInformation: /Core/PresenceType[0..1]
EnergyConsumption
+ finetuningEnergyConsumption: EnergyConsumptionDescription[0..*]
+ inferenceEnergyConsumption: EnergyConsumptionDescription[0..*]
+ trainingEnergyConsumption: EnergyConsumptionDescription[0..*]
EnergyConsumptionDescription
+ energyQuantity: xsd:decimal[1]
+ energyUnit: EnergyUnitType[1]
Enumerations
EnergyUnitType
SafetyRiskAssessmentType

Figure A.6: AI Profile

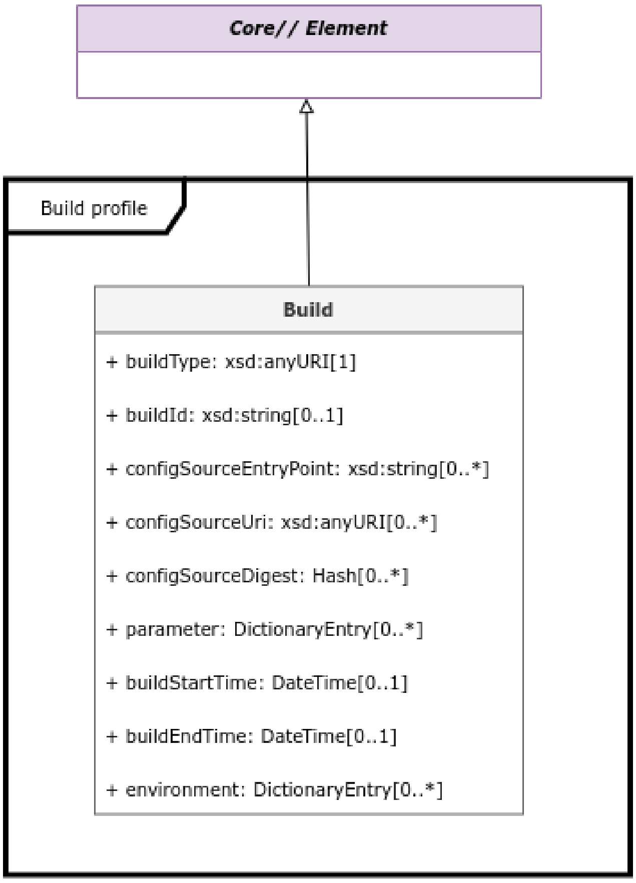
Core// Element
Build profile
Build
+ buildType: xsd:anyURI[1]
+ buildId: xsd:string[0..1]
+ configSourceEntryPoint: xsd:string[0..*]
+ configSourceUri: xsd:anyURI[0..*]
+ configSourceDigest: Hash[0..*]
+ parameter: DictionaryEntry[0..*]
+ buildStartTime: DateTime[0..1]
+ buildEndTime: DateTime[0..1]
+ environment: DictionaryEntry[0..*]

Figure A.7: Build Profile

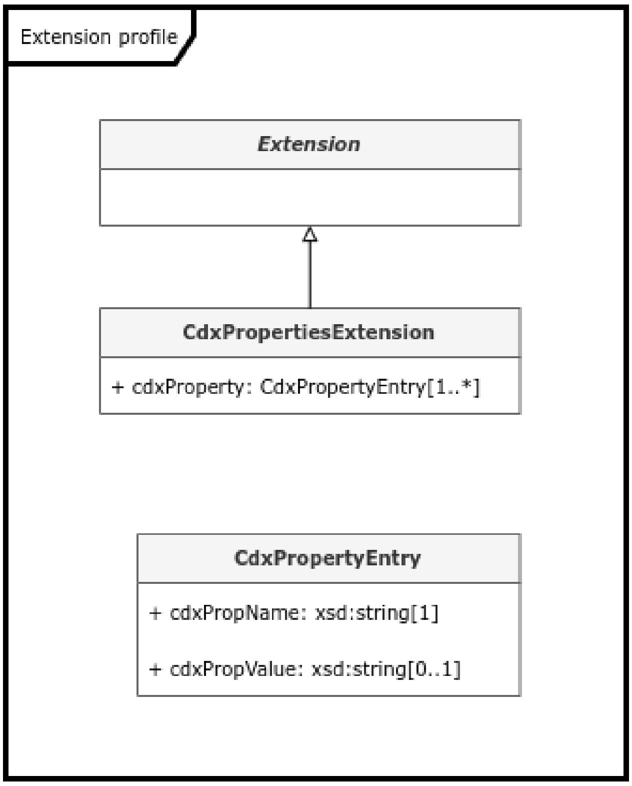
Extension profile
Extension
CdxPropertiesExtension
+ cdxProperty: CdxPropertyEntry[1..*]
CdxPropertyEntry
+ cdxPropName: xsd:string[1]
+ cdxPropValue: xsd:string[0..1]

Figure A.8: Extension Profile

Annex B

SPDX license expressions (Normative)

B.1 Overview

Often a single license can be used to represent the licensing terms of a source code or binary file, but there are situations where a single license identifier is not sufficient. A common example is when software is offered under a choice of one or more licenses (e.g., GPL-2.0-only OR BSD-3-Clause). Another example is when a set of licenses is needed to represent a binary program constructed by compiling and linking two (or more) different source files each governed by different licenses (e.g., LGPL-2.1-only AND BSD-3-Clause).

SPDX License Expressions provide a way for one to construct expressions that more accurately represent the licensing terms typically found in open source software source code. A license expression could be a single license identifier found on the SPDX License List; a user defined license reference denoted by the LicenseRef-`[idString]`; a license identifier combined with an SPDX exception; or some combination of license identifiers, license references and exceptions constructed using a small set of defined operators (e.g., `AND`, `OR`, `WITH` and `+`). We provide the definition of what constitutes a valid SPDX License Expression in this section.

The exact syntax of license expressions is described below in ABNF, as defined in RFC 5234[1] and expanded in RFC 7405[2].

```
idstring = 1*(ALPHA / DIGIT / "-" / "." )

license-id = <short form license identifier from SPDX License List>

license-exception-id = <short form license exception identifier from SPDX License List>

license-ref = [%s"DocumentRef-"(idstring)":"]%s"LicenseRef-"(idstring)

addition-ref = [%s"DocumentRef-"(idstring)":"]%s"AdditionRef-"(idstring)

simple-expression = license-id / license-id"+" / license-ref

addition-expression = license-exception-id / addition-ref

compound-expression = (simple-expression /

  simple-expression ( %s"WITH" / %s"with" ) addition-expression /

  compound-expression ( %s"AND" / %s"and" ) compound-expression /

  compound-expression ( %s"OR" / %s"or" ) compound-expression /

  "(" compound-expression ")" )

license-expression = (simple-expression / compound-expression)
```

In the following sections we describe in more detail `<license-expression>` construct, a licensing expression string that enables a more accurate representation of the licensing terms of modern-day software.

A valid `<license-expression>` string consists of either:

(i) a simple license expression, such as a single license identifier; or

(ii) a more complex expression constructed by combining smaller valid expressions using Boolean license operators.

[1] https://datatracker.ietf.org/doc/rfc5234/

[2] https://datatracker.ietf.org/doc/rfc7405/

There MUST NOT be white space between a license-id and any following +. This supports easy parsing and backwards compatibility. There MUST be white space on either side of the operator "WITH". There MUST be white space and/or parentheses on either side of the operators AND and OR.

In the tag:value format, a license expression MUST be on a single line, and MUST NOT include a line break in the middle of the expression.

B.2 Case sensitivity

License expression operators (AND, and, OR, or, WITH and with) should be matched in a *case-sensitive* manner, i.e., letters must be all upper case or all lower case.

License identifiers (including license exception identifiers) used in SPDX documents or source code files should be matched in a *case-insensitive* manner. In other words, MIT, Mit and mIt should all be treated as the same identifier and referring to the same license.

However, please be aware that it is often important to match with the case of the canonical identifier on the SPDX License List[3]. This is because the canonical identifier's case is used in the URL of the license's or exception's entry on the List, and because the canonical identifier is translated to a URI in RDF documents.

For user defined license identifiers, only the variable part (after LicenseRef-) is case insensitive. This means, for example, that LicenseRef-Name and LicenseRef-name should be treated as the same identifier and considered to refer to the same license, while licenseref-name is not a valid license identifier.

The same applies to AdditionRef- user defined identifiers.

B.3 Simple license expressions

A simple <license-expression> is composed one of the following:

- An SPDX License List Short Form Identifier. For example: CDDL-1.0
- An SPDX License List Short Form Identifier with a unary "+" operator suffix to represent the current version of the license or any later version. For example: CDDL-1.0+
- An SPDX user defined license reference: ["DocumentRef-"1*(idstring)":"]"LicenseRef-"1*(idstring)

Some examples:

```
LicenseRef-23
```

```
LicenseRef-MIT-Style-1
```

```
DocumentRef-spdx-tool-1.2:LicenseRef-MIT-Style-2
```

The current set of valid license identifiers can be found in spdx.org/licenses[4].

B.4 Composite license expressions

B.4.1 Introduction

More expressive composite license expressions can be constructed using "OR", "AND", and "WITH" operators similar to constructing mathematical expressions using arithmetic operators.

For the tag:value format, any license expression that consists of more than one license identifier and/or LicenseRef, may optionally be encapsulated by parentheses: "( )".

Nested parentheses can also be used to specify an order of precedence which is discussed in more detail in Order of precedence and parentheses.

[3] https://spdx.org/licenses

[4] https://spdx.org/licenses

B.4.2 Disjunctive “OR” operator

If presented with a choice between two or more licenses, use the disjunctive binary “OR” operator to construct a new license expression, where both the left and right operands are valid license expression values.

For example, when given a choice between the LGPL-2.1-only or MIT licenses, a valid expression would be:

```
LGPL-2.1-only OR MIT
```

The “OR” operator is commutative, meaning that the above expression should be considered equivalent to:

```
MIT OR LGPL-2.1-only
```

An example representing a choice between three different licenses would be:

```
LGPL-2.1-only OR MIT OR BSD-3-Clause
```

It is allowed to use the operator in lower case form `or`.

B.4.3 Conjunctive “AND” operator

If required to simultaneously comply with two or more licenses, use the conjunctive binary “AND” operator to construct a new license expression, where both the left and right operands are a valid license expression values.

For example, when one is required to comply with both the LGPL-2.1-only or MIT licenses, a valid expression would be:

```
LGPL-2.1-only AND MIT
```

The “AND” operator is commutative, meaning that the above expression should be considered equivalent to:

```
MIT AND LGPL-2.1-only
```

An example where all three different licenses apply would be:

```
LGPL-2.1-only AND MIT AND BSD-2-Clause
```

It is allowed to use the operator in lower case form `and`.

B.4.4 Additive “WITH” operator

Sometimes license texts are found with additional text, which might or might not modify the original license terms.

In this case, use the binary “WITH” operator to construct a new license expression to represent the special situation. A valid `<license-expression>` is where the left operand is a `<simple-expression>` value and the right operand is a `<addition-expression>` that represents the additional text.

The `<addition-expression>` can be either a `<license-exception-id>` from the SPDX License List, or a user defined addition reference in the form [“DocumentRef-”(idstring)“:”]“AdditionRef-”(idstring)

For example, when the Bison exception is to be applied to GPL-2.0-or-later, the expression would be:

```
GPL-2.0-or-later WITH Bison-exception-2.2
```

The current set of valid license exceptions identifiers can be found in spdx.org/licenses[5].

It is allowed to use the operator in lower case form `with`.

[5] https://spdx.org/licenses

B.4.5 Order of precedence and parentheses

The order of application of the operators in an expression matters (similar to mathematical operators). The default operator order of precedence of a `<license-expression>` a is:

```
+
WITH
AND
OR
```

where a lower order operator is applied before a higher order operator.

For example, the following expression:

```
LGPL-2.1-only OR BSD-3-Clause AND MIT
```

represents a license choice between either LGPL-2.1-only and the expression BSD-3-Clause AND MIT because the AND operator takes precedence over (is applied before) the OR operator.

When required to express an order of precedence that is different from the default order a `<license-expression>` can be encapsulated in pairs of parentheses: ( ), to indicate that the operators found inside the parentheses takes precedence over operators outside. This is also similar to the use of parentheses in an algebraic expression e.g., (5+7)/2.

For instance, the following expression:

```
MIT AND (LGPL-2.1-or-later OR BSD-3-Clause)
```

states the OR operator should be applied before the AND operator. That is, one should first select between the LGPL-2.1-or-later or the BSD-3-Clause license before applying the MIT license.

B.4.6 License expressions in RDF

A conjunctive license can be expressed in RDF via a `<spdx:ConjunctiveLicenseSet>` element, with an spdx:member property for each element in the conjunctive license. Two or more members are required.

```
<spdx:ConjunctiveLicenseSet>
    <spdx:member rdf:resource="http://spdx.org/licenses/GPL-2.0-only"/>
    <spdx:ExtractedLicensingInfo rdf:about
      ="http://example.org#LicenseRef-EternalSurrender">
        <spdx:extractedText>
            In exchange for using this software, you agree to give
            its author all your worldly possessions. You will not
            hold the author liable for all the damage this software
            will inevitably cause not only to your person and
            property, but to the entire fabric of the cosmos.
        </spdx:extractedText>
        <spdx:licenseId>LicenseRef-EternalSurrender</spdx:licenseId>
    </spdx:ExtractedLicensingInfo>
</spdx:ConjunctiveLicenseSet>
```

A disjunctive license can be expressed in RDF via a `<spdx:DisjunctiveLicenseSet>` element, with an spdx:member property for each element in the disjunctive license. Two or more members are required.

```
<spdx:DisjunctiveLicenseSet>
    <spdx:member rdf:resource="http://spdx.org/licenses/GPL-2.0-only"/>
    <spdx:member>
        <spdx:ExtractedLicensingInfo rdf:about
          ="http://example.org#LicenseRef-EternalSurrender">
            <spdx:extractedText>
                In exchange for using this software, you agree to
                give its author all your worldly possessions. You
                will not hold the author liable for all the damage
                this software will inevitably cause not only to
                your person and property, but to the entire fabric
                of the cosmos.
            </spdx:extractedText>
            <spdx:licenseId>LicenseRef-EternalSurrender</spdx:licenseId>
        </spdx:ExtractedLicensingInfo>
    </spdx:member>
</spdx:DisjunctiveLicenseSet>
```

A License Exception can be expressed in RDF via a `<spdx:LicenseException>` element. This element has the following unique mandatory (unless specified otherwise) attributes:

- `comment` - An `rdfs:comment` element describing the nature of the exception.
- `seeAlso` (optional, one or more)- An `rdfs:seeAlso` element referencing external sources of information on the exception.
- `example` (optional) - Text describing examples of this exception.
- `name` - The full human readable name of the item.
- `licenseExceptionId` - The identifier of an exception in the SPDX License List to which the exception applies.
- `licenseExceptionText` - Full text of the license exception.

```
<rdf:Description rdf:about
  ="http://example.org#SPDXRef-ButIdDontWantToException">
    <rdfs:comment>This exception may be invalid in some
      jurisdictions.</rdfs:comment>
    <rdfs:seeAlso>http://dilbert.com/strip/1997-01-15</rdfs:seeAlso>
    <spdx:example>So this one time, I had a license exception
      …</spdx:example>
    <spdx:licenseExceptionText>
        A user of this software may decline to follow any subset of
        the terms of this license upon finding any or all such terms
        unfavorable.
    </spdx:licenseExceptionText>
    <spdx:name>&quot;But I Don&apos;t Want To&quot; Exception</spdx:name>
    <spdx:licenseExceptionId>SPDXRef-ButIdDontWantToException</spdx:licenseExceptionId>
    <rdf:type rdf:resource
      ="http://spdx.org/rdf/terms#LicenseException"/>
</rdf:Description>
```

Annex C

SPDX License List matching guidelines and templates (Normative)

C.1 SPDX License List matching guidelines

The SPDX License List Matching Guidelines provide guidelines to be used for the purposes of matching licenses and license exceptions against those included on the SPDX License List[1]. There is no intent here to make a judgment or interpretation, but merely to ensure that when one SPDX user identifies a license as "BSD-3-Clause," for example, it is indeed the same license as what someone else identifies as "BSD-3-Clause" and the same license as what is listed on the SPDX License List. As noted here, some of the matching guidelines are implemented in the XML files of the SPDX License List repository.

C.2 How these guidelines are applied

C.2.1 Purpose

To ensure consistent results by different SPDX document creators when matching license information that will be included in SPDX data. SPDX document creators or tools may match on the license or exception text itself, the official license header, or the SPDX License List short identifier.

C.2.2 Guideline: official license headers

The matching guidelines apply to license and exception text, as well as official license headers. Official license headers are defined by the SPDX License List as specific text specified within the license itself to be put in the header of files. (see explanation of SPDX License List fields[2] for more info).

The following XML tag is used to implement this guideline: `<standardLicenseHeader>`

C.3 Substantive text

C.3.1 Purpose

To ensure that when matching licenses and exceptions to the SPDX License List, there is an appropriate balance between matching against the substantive text and disregarding parts of the text that do not alter the substantive text or legal meaning. Further guidelines of what can be disregarded or considered replaceable for purposes of matching are listed below here and in the subsequent specific guidelines. A conservative approach is taken in regard to rules relating to disregarded or replaceable text.

C.3.2 Guideline: verbatim text

License and exception text should be the same verbatim text (except for the guidelines stated here). The text should be in the same order, e.g., differently ordered paragraphs would not be considered a match.

C.3.3 Guideline: no additional text

Matched text should only include that found in the vetted license or exception text. Where a license or exception found includes additional text or clauses, this should not be considered a match.

C.3.4 Guideline: replaceable text

Some licenses include text that refers to the specific copyright holder or author, yet the rest of the license is exactly the same. The intent here is to avoid the inclusion of a specific name in one part of the license resulting in a non-match where the license is otherwise an exact match to the legally substantive terms (e.g., the third clause and disclaimer in the BSD licenses, or the third, fourth, and fifth clauses of Apache-1.1). In these cases, there should be a positive license match.

[1] https://spdx.org/licenses/

[2] https://github.com/spdx/license-list-XML/blob/v3.25.0/DOCS/license-fields.md

The text indicated as such can be replaced with similar values (e.g., a different name or generic term; different date) and still be considered a positive match. This rule also applies to text-matching in official license headers, see Guideline: official license headers.

The following XML tag is used to implement this guideline. `<alt>` with 2 attributes:

- `match` - a POSIX extended regular expression (ERE) to match the replaceable text
- `name` - an identifier for the variable text unique to the license XML document

The original text is enclosed within the beginning and ending alt tags.

For example: `<alt match="(?i:copyright.{0,200})." name="copyright1">Copyright The Linux Foundation</alt>`

The original replaceable text appears on the SPDX License List webpage in red text.

C.3.5 Guideline: omittable text

Some licenses have text that can simply be ignored. The intent here is to avoid the inclusion of certain text that is superfluous or irrelevant in regards to the substantive license text resulting in a non-match where the license is otherwise an exact match (e.g., directions on how to apply the license or other similar exhibits). In these cases, there should be a positive license match.

The license should be considered a match if the text indicated is present and matches OR the text indicated is missing altogether.

The following XML tag is used to implement this guideline: `<optional>`

For example: `<optional>Apache License Version 2.0, January 2004 http://www.apache.org/licenses/<`

Omittable text appears on the SPDX License List webpage in blue text.

C.4 Whitespace

C.4.1 Purpose

To avoid the possibility of a non-match due to different spacing of words, line breaks, or paragraphs.

C.4.2 Guideline

All whitespace should be treated as a single blank space.

XML files do not require specific markup to implement this guideline.

C.5 Capitalization

C.5.1 Purpose

To avoid the possibility of a non-match due to lowercase or uppercase letters in otherwise the same words.

C.5.2 Guideline

All uppercase and lowercase letters should be treated as lowercase letters.

XML files do not require specific markup to implement this guideline.

C.6 Punctuation

C.6.1 Purpose

Because punctuation can change the meaning of a sentence, punctuation needs to be included in the matching process.

XML files do not require specific markup to implement this guideline, unless to indicate an exception to the guideline.

C.6.2 Guideline: punctuation

Punctuation should be matched, unless otherwise stated in these guidelines or unless specific markup is added.

C.6.3 Guideline: hyphens, dashes

Any hyphen, dash, en dash, em dash, or other variation should be considered equivalent.

C.6.4 Guideline: Quotes

Any variation of quotations (single, double, curly, etc.) should be considered equivalent.

C.7 Code Comment Indicators or Separators

C.7.1 Purpose

To avoid the possibility of a non-match due to the existence or absence of code comment indicators placed within the license text, e.g., at the start of each line of text, or repetitive characters to establish a separation of text, e.g., `---`, ===, ____, or `***`.

C.7.2 Guideline

Any kind of code comment indicator or prefix which occurs at the beginning of each line in a matchable section should be ignored for matching purposes.

XML files do not require specific markup to implement this guideline.

C.7.3 Guideline

A non-letter character repeated 3 or more times to establish a visual separation should be ignored for matching purposes.

XML files do not require specific markup to implement this guideline.

C.8 Bullets and numbering

C.8.1 Purpose

To avoid the possibility of a non-match due to the otherwise same license using bullets instead of numbers, number instead of letter, or no bullets instead of bullet, etc., for a list of clauses.

C.8.2 Guideline

Where a line starts with a bullet, number, letter, or some form of a list item (determined where list item is followed by a space, then the text of the sentence), ignore the list item for matching purposes.

The following XML tag is used to implement this guideline: `<bullet>`

For example: `<bullet>1.0</bullet>`

C.9 Varietal word spelling

C.9.1 Purpose

English uses different spelling for some words. By identifying the spelling variations for words found or likely to be found in licenses, we avoid the possibility of a non-match due to the same word being spelled differently. This list is not meant to be an exhaustive list of all spelling variations, but meant to capture the words most likely to be found in open source software licenses.

C.9.2 Guideline

The words in each line of the text file available at the equivalent words list[3] are considered equivalent and interchangeable.

XML files do not require specific markup to implement this guideline.

C.10 Copyright symbol

C.10.1 Purpose

By having a rule regarding the use of “©”, “(c)”, or “copyright”, we avoid the possibility of a mismatch based on these variations.

[3] https://spdx.org/licenses/equivalentwords.txt

C.10.2 Guideline

"©", "(c)", or "Copyright" should be considered equivalent and interchangeable.

XML files do not require specific markup to implement this guideline. The copyright symbol is part of the copyright notice, see implementation of that guideline in Copyright notice.

C.11 Copyright notice

C.11.1 Purpose

To avoid a license mismatch merely because the copyright notice (usually found above the actual license or exception text) is different. The copyright notice is important information to be recorded elsewhere in the SPDX document, but for the purposes of matching a license to the SPDX License List, it should be ignored because it is not part of the substantive license text.

C.11.2 Guideline

Ignore copyright notices. A copyright notice consists of the following elements, for example: "2012 Copyright, John Doe. All rights reserved." or "(c) 2012 John Doe."

The following XML tag is used to implement this guideline: `<copyrightText>`

For example: `<copyrightText>Copyright 2022 The Linux Foundation</copyrightText>`

C.12 License name or title

C.12.1 Purpose

To avoid a license mismatch merely because the name or title of the license is different than how the license is usually referred to or different than the SPDX full name. This also avoids a mismatch if the title or name of the license is simply not included.

C.12.2 Guideline

Ignore the license name or title for matching purposes, so long as what ignored is the title only and there is no additional substantive text added here.

The following XML tag is used to implement this guideline: `<titleText>`

For example: `<titleText>Attribution Assurance License</titleText>`

C.13 Extraneous text at the end of a license

C.13.1 Purpose

To avoid a license mismatch merely because extraneous text that appears at the end of the terms of a license is different or missing. This also avoids a mismatch if the extraneous text merely serves as a license notice example and includes a specific copyright holder's name.

C.13.2 Guideline

Ignore any text that occurs after the obvious end of the license and does not include substantive text of the license, for example: text that occurs after a statement such as, "END OF TERMS AND CONDITIONS," or an exhibit or appendix that includes an example or instructions on to how to apply the license to your code. Do not apply this guideline or ignore text that is comprised of additional license terms (e.g., permitted additional terms under GPL-3.0, section 7).

To implement this guideline, use the `<optional>` XML element tag as described in Guideline: omittable text.

C.14 HTTP protocol

C.14.1 Purpose

To avoid a license mismatch due to a difference in a hyperlink protocol (e.g. HTTP vs. HTTPS).

C.14.2 Guideline

`http://` and `https://` should be considered equivalent.

XML files do not require specific markup to implement this guideline.

C.15 SPDX License List

C.15.1 Template access

The license XML can be accessed in the license-list-data repository under the license-list-XML directory. Although the license list XML files can also be found in the license-list-XML[4] repository, users are encouraged to use the published versions in the license-list-data[5] repository. The license-list-data repository is tagged by release. Only tagged released versions of the license list are considered stable.

C.15.2 License List XML format

A full schema for the License List XML can be found at SPDX License List XML Schema[6].

C.15.3 Legacy Text Template format

Prior to the XML format, a text template was used to express variable and optional text in licenses. This text template is still supported, however, users are encouraged to use the more expressive XML format.

A legacy template is composed of text with zero or more rules embedded in it.

A rule is a variable section of a license wrapped between double angle brackets <<>> and is composed of 4 fields. Each field is separated with a semi-colon `;`. Rules cannot be embedded within other rules. Rule fields begin with a case sensitive tag followed by an equal sign =.

Rule fields:

- **type:** indicates whether the text is replaceable or omittable as per Substantive text guidelines.
  - Indicated by `<<var; . . . >>` or
  - Indicated by `<<beginOptional; . . .>>` and `<<endOptional>>` respectively.
  - This field is the first field and is required.
- **name:** name of the field in the template.
  - This field is unique within each license template.
  - This field is required.
- **original:** the original text of the rule.
  - This field is required for a rule type: `<<var; . . . >>`
- **match:** a POSIX extended regular expression (ERE).
  - This field is required for a rule type: `<<var; . . . >>`

The POSIX ERE[7] in the match field has the following restrictions and extensions:

- Semicolons are escaped with `\;`
- POSIX Bracket Extensions are not allowed

For example: `<<var;name=organizationClause3;original=the copyright holder;match=.+>>`

[4] https://github.com/spdx/license-list-XML

[5] https://github.com/spdx/license-list-data

[6] https://github.com/spdx/license-list-XML/blob/v3.25.0/schema/ListedLicense.xsd

[7] http://pubs.opengroup.org/onlinepubs/9699919799/

Annex D

SPDX Lite (Normative)

D.1 Explanation of the Lite profile

The Lite profile is designed to make it quick and easy to start a Software Bill of Materials in situations where a company may have limited capacity for introducing new items into their processes. The Lite profile captures the minimum set of information required for license compliance in the software supply chain. It contains information about the creation of the SBOM, package lists with licensing and other related information, and their relationships.

All elements in Lite profile are essential for complying with licenses. It is easy to use a SPDX document with the Lite profile for anyone who does not have enough knowledge about licensing information and easy to import license information from former versions of SPDX Lite format files. The Lite profile offers the flexibility to be used either alone or in combination with other SPDX profiles as a SPDX document in the software supply chain.

D.2 Mandatory and recommended properties

The Lite profile specifies that some properties MUST be present and some others SHOULD be present, as much as possible.

The following lists collect and present this information for every class present in the SPDX data, in a concise and easy-to-follow format. The lists of properties are in alphabetical order, for easy reference.

D.2.1 /Core/SpdxDocument

- Mandatory

    1. creationInfo
    2. element (may be multiple), MUST have at least one /Core/Sbom object
    3. rootElement (may be multiple), SHOULD be objects of type /Core/Sbom
    4. spdxId

- Recommended

    1. comment
    2. dataLicense
    3. name
    4. namespaceMap (may be multiple)
    5. verifiedUsing (may be multiple), SHOULD be objects of type /Core/Hash

D.2.2 /Software/Sbom

- Mandatory

    1. creationInfo
    2. element (may be multiple), MUST have at least one /Software/Package object
    3. rootElement (may be multiple), SHOULD be objects of type /Software/Package
    4. spdxId

- Recommended

    1. sbomType (may be multiple)

D.2.3 /Software/Package

- Mandatory

    1. copyrightText
    2. creationInfo
    3. name
    4. packageVersion

5. spdxId
6. suppliedBy, SHOULD be an object of type /Core/Agent

- Recommended

  1. attributionText (may be multiple)
  2. builtTime
  3. comment
  4. downloadLocation
  5. homepage
  6. originatedBy (may be multiple), SHOULD be objects of type /Core/Agent
  7. packageUrl
  8. releaseTime
  9. supportLevel (may be multiple)
  10. validUntilTime
  11. verifiedUsing (may be multiple), SHOULD be objects of type /Core/Hash

However, there MUST be at least a “downloadLocation” or “packageUrl” property.

Additionally:

1. for every /Software/Package object MUST exist exactly one /Core/Relationship object of type `hasConcludedLicense` having that element as its `from` property and an /SimpleLicensing/AnyLicenseInfo as its `to` property.
2. for every /Software/Package object MUST exist exactly one /Core/Relationship object of type `hasDeclaredLicense` having that element as its `from` property and /SimpleLicensing/AnyLicenseInfo object as its `to` property.

D.2.4 /Core/Hash

- Mandatory

  1. algorithm
  2. hashValue

- Recommended

  1. comment

D.2.5 /SimpleLicensing/LicenseExpression

- Mandatory

  1. creationInfo
  2. licenseExpression
  3. spdxId

- Recommended

  1. licenseListVersion

D.2.6 /SimpleLicensing/SimpleLicensingText

- Mandatory

  1. creationInfo
  2. licenseText
  3. spdxId

- Recommended

  1. comment

D.2.7 /Core/Agent (createdBy, suppliedBy, originatedBy)

- Mandatory

  1. creationInfo, SHOULD be “BlankNode”
  2. name
  3. spdxId

- Recommended

  1. externalIdentifier (may be multiple)

D.2.8 /Core/CreationInfo

- Mandatory

  1. created
  2. createdBy (may be multiple), SHOULD be objects of type /Core/Agent
  3. specVersion, MUST be a fixed string, “3.0.1”.

- Recommended

  1. comment

D.2.9 /Core/ExternalIdentifier

- Mandatory

  1. externalIdentifierType
  2. identifier

D.2.10 /Core/NameSpaceMap

- Mandatory

  1. namespace
  2. prefix

D.2.11 /Core/Relationship

- Mandatory

  1. creationInfo
  2. from
  3. relationshipType
  4. spdxId
  5. to (may be multiple)

Annex E

Package URL specification v1 (Normative)

E.1 Introduction

The Package URL core specification defines a versioned and formalized format, syntax, and rules used to represent and validate package URLs.

A package URL or *purl* is an attempt to standardize existing approaches to reliably identify the location of software packages.

A *purl* is a URL string used to identify the location of a software package in a mostly universal and uniform way across programming languages, package managers, packaging conventions, tools, APIs and databases.

Such a package URL is useful to reliably reference the same software package using a simple and expressive syntax and conventions based on familiar URLs.

E.2 Syntax definition

purl stands for **package URL**.

A *purl* is a URL composed of seven components:

```
scheme:type/namespace/name@version?qualifiers#subpath
```

Components are separated by a specific character for unambiguous parsing.

The definition for each components is:

- **scheme**: this is the URL scheme with the constant value of “`pkg`”. One of the primary reason for this single scheme is to facilitate the future official registration of the “`pkg`” scheme for package URLs. Required.
- **type**: the package type or package protocol such as maven, npm, nuget, gem, pypi, etc. Required.
- **namespace**: some name prefix such as a Maven groupid, a Docker image owner, a GitHub user or organization. Optional and type-specific.
- **name**: the name of the package. Required.
- **version**: the version of the package. Optional.
- **qualifiers**: extra qualifying data for a package such as an OS, architecture, a distribution, etc. Optional and type-specific.
- **subpath**: extra subpath within a package, relative to the package root. Optional.

Components are designed such that they form a hierarchy from the most significant on the left to the least significant components on the right.

A *purl* is a valid URL and URI that conforms to the URL definitions and specifications in RFC 3986 https://datatracker.ietf.org/doc/rfc3986.

A *purl* must not contain a URL Authority i.e. there is no support for username, password, host and port components. A `namespace` segment may sometimes look like a host but its interpretation is specific to a type.

The *purl* components are mapped to the following URL components:

- *purl* scheme: this is a URL scheme with a constant value: `pkg`
- *purl* type, namespace, name and version components: these are collectively mapped to a URL path
- *purl* qualifiers: this maps to a URL query
- *purl* subpath: this is a URL fragment

E.3 Character encoding

For clarity and simplicity a *purl* is always an ASCII string. To ensure that there is no ambiguity when parsing a *purl*, separator characters and non-ASCII characters must be encoded in UTF-8, and then percent-encoded as defined in RFC 3986 https://datatracker.ietf.org/doc/rfc3986.

Use these rules for percent-encoding and decoding *purl* components:

- the type must NOT be encoded and must NOT contain separators
- the `#`, `?`, `@` and `:` characters must NOT be encoded when used as separators. They may need to be encoded elsewhere
- the `:` scheme and type separator does not need to and must NOT be encoded. It is unambiguous unencoded everywhere
- the `/` used as type/namespace/name and subpath segments separator does not need to and must NOT be percent-encoded. It is unambiguous unencoded everywhere
- the `@` version separator must be encoded as `%40` elsewhere
- the `?` qualifiers separator must be encoded as `%3F` elsewhere
- the = qualifiers key/value separator must NOT be encoded
- the `#` subpath separator must be encoded as `%23` elsewhere
- All non-ASCII characters must be encoded as UTF-8 and then percent-encoded

It is OK to percent-encode any *purl* components, except for the type. Producers and consumers of *purl* data must always percent-decode and percent-encode components and component segments as explained in the "How to produce and consume *purl* data" section.

E.4 Rules for each component

A *purl* string is an ASCII URL string composed of seven components.

Some components are allowed to use other characters beyond ASCII: these components must then be UTF-8-encoded strings and percent-encoded as defined in the "Character encoding" section.

The rules for each component are:

E.4.1 Rules for scheme

- The scheme is a constant with the value "`pkg`"
- Since a *purl* never contains a URL Authority, its scheme must not be suffixed with double slash as in `pkg://` and should use instead `pkg:`.
- *purl* parsers must accept URLs such as 'pkg://' and must ignore the '//'.
- *purl* builders must not create invalid URLs with such double slash '//'.
- The scheme is followed by a ':' separator.
- For example, the two purls `pkg:gem/ruby-advisory-db-check@0.12.4` and `pkg://gem/ruby-advisory-db-chec` are strictly equivalent. The first is in canonical form while the second is an acceptable *purl* but is an invalid URI/URL per RFC3986.

E.4.2 Rules for type

- The package type is composed only of ASCII letters and numbers, `.`, `+` and `-` (period, plus, and dash).
- The type cannot start with a number.
- The type cannot contain spaces.
- The type must not be percent-encoded.
- The type is case insensitive, with the canonical form being lowercase.

E.4.3 Rules for namespace

- The optional namespace contains zero or more segments, separated by slash `/`.
- Leading and trailing slashes `/` are not significant and should be stripped in the canonical form. They are not part of the namespace.
- Each namespace segment must be a percent-encoded string.
- When percent-decoded, a segment must not contain a slash `/` and must not be empty.
- A URL host or Authority must NOT be used as a namespace. Use instead a `repository_url` qualifier. Note however that for some types, the namespace may look like a host.

E.4.4 Rules for name

- The name is prefixed by a slash `/` separator when the namespace is not empty.
- This slash `/` is not part of the name.
- A name must be a percent-encoded string.

E.4.5 Rules for version

- The version is prefixed by a at-sign `@` separator when not empty.
- This at-sign `@` is not part of the version.
- A version must be a percent-encoded string.
- A version is a plain and opaque string. Some package types use versioning conventions such as SemVer for NPMs or NEVRA conventions for RPMS. A type may define a procedure to compare and sort versions, but there is no reliable and uniform way to do such comparison consistently.

E.4.6 Rules for qualifiers

- The qualifiers string is prefixed by a `?` separator when not empty.
- This `?` is not part of the qualifiers.
- This is a string composed of zero or more key=value pairs each separated by an ampersand `&`. A key and value are separated by an equal = character.
- These `&` are not part of the key=value pairs.
- Each key must be unique within the keys of the qualifiers string.
- A value cannot be an empty string; a key=value pair with an empty value is the same as no key/value at all for this key.
- Each key must be composed only of ASCII letters and numbers, `.`, `-` and _ (period, dash and underscore).
- A key cannot start with a number.
- A key must NOT be percent-encoded.
- A key is case insensitive, with the canonical form being lowercase.
- A key cannot contain spaces.
- A value must be a percent-encoded string.
- The = separator is neither part of the key nor of the value.

E.4.7 Rules for subpath

- The subpath string is prefixed by a `#` separator when not empty.
- This `#` is not part of the subpath.
- The subpath contains zero or more segments, separated by slash `/`.
- Leading and trailing slashes `/` are not significant and should be stripped in the canonical form.
- Each subpath segment must be a percent-encoded string.
- When percent-decoded, a segment must not contain a `/`, must not be any of `..` or `.`, and must not be empty.
- The subpath must be interpreted as relative to the root of the package.

E.5 Known types

There are several known *purl* package type definitions. The current list of known types is: `alpm`, `apk`, `bitbucket`, `bitnami`, `cargo`, `cocoapods`, `composer`, `conan`, `conda`, `cpan`, `cran`, `deb`, `docker`, `gem`, `generic`, `github`, `golang`, `hackage`, `hex`, `huggingface`, `luarocks`, `maven`, `mlflow`, `npm`, `nuget`, `oci`, `pub`, `pypi`, `qpkg`, `rpm`, `swid`, and `swift`.

The list, with definitions for each type, is maintained in the file named `PURL-TYPES.rst` in the online repository https://github.com/package-url/purl-spec.

E.6 Known qualifiers key/value pairs

Qualifiers should be limited to the bare minimum for proper package identification, to ensure that a *purl* stays compact and readable in most cases. Separate external attributes stored outside of a *purl* are the preferred mechanism to convey extra long and optional information. API, database or web form.

The following keys are valid for use in all package types:

- `repository_url` is an extra URL for an alternative, non-default package repository or registry. The default repository or registry of each type is documented in the “Known types” section.

- `download_url` is an extra URL for a direct package web download URL.
- `vcs_url` is an extra URL for a package version control system URL.
- `file_name` is an extra file name of a package archive.
- `checksum` is a qualifier for one or more checksums stored as a comma-separated list. Each item in the list is in form of algorithm:hex_value (all lowercase), such as `sha1:ad9503c3e994a4f611a4892f2e67ac82df727086`.

E.7 How to produce and consume *purl* data

The following provides rules to be followed when building or deconstructing *purl* instances.

E.7.1 How to build *purl* string from its components

Building a *purl* ASCII string works from left to right, from type to subpath.

To build a *purl* string from its components:

1. Start a *purl* string with the "`pkg:`" scheme as a lowercase ASCII string
2. Append the type string to the *purl* as a lowercase ASCII string
3. Append `/` to the *purl*
4. If the namespace is not empty:
    1. Strip the namespace from leading and trailing `/`
    2. Split on `/` as segments
    3. Apply type-specific normalization to each segment, if needed
    4. Encode each segment in UTF-8-encoding
    5. Percent-encode each segment
    6. Join the segments with `/`
    7. Append this to the *purl*
    8. Append `/` to the *purl*
5. Strip the name from leading and trailing `/`
6. Apply type-specific normalization to the name, if needed
7. Encode the name in UTF-8-encoding
8. Percent-encode the name
9. Append the percent-encoded name to the *purl*
10. If the version is not empty:
    1. Append `@` to the *purl*
    2. Encode the version in UTF-8-encoding
    3. Percent-encode the version
    4. Append the percent-encoded version to the *purl*
11. If the qualifiers are not empty and not composed only of key/value pairs where the value is empty:
    1. Append `?` to the *purl*
    2. Discard any pair where the value is empty
    3. Encode each value in UTF-8-encoding
    4. If the key is `checksum` and there are more than one checksums, join the list with `,` to create the qualifier value
    5. Create each qualifier string by joining the lowercased key, the equal = sign, and the percent-encoded value
    6. Sort this list of qualifier strings lexicographically
    7. Join this list of sorted qualifier strings with `&`
    8. Append this string to the *purl*
12. If the subpath is not empty and not composed only of empty, `.`, and `..` segments:
    1. Append `#` to the *purl*
    2. Strip the subpath from leading and trailing `/`

3. Split the subpath on `/` as a list of segments
4. Discard empty, `.`, and `..` segments
5. Encode each segment in UTF-8-encoding
6. Percent-encode each segment
7. Join the segments with `/`
8. Append this string to the *purl*

E.7.2 How to parse a *purl* string to its components

Parsing a *purl* ASCII string into its components works by splitting the string on different characters.

To parse a *purl* string in its components:

1. Split the *purl* string once from right on `#`, if present; the left side is the remainder.
2. If the right side is not empty, it contains subpath information:
    1. Strip it from leading and trailing `/`.
    2. Split this on `/` in a list of segments.
    3. Discard empty, `.`, and `..` segments.
    4. Percent-decode each segment.
    5. UTF-8-decode each of these.
    6. Join segments with `/`.
    7. This is the subpath.
3. Split the remainder once from right on `?`, if present; the left side is the remainder.
4. If the right side is not empty, it contains qualifiers information:
    1. Split it on `&` in a list of key=value pairs.
    2. Split each pair once from left on = in key and value parts.
    3. The key is the lowercase left side.
    4. Percent-decode the right side.
    5. UTF-8-decode this to get the value.
    6. Discard any key/value pairs where the value is empty.
    7. If the key is `checksum`, split the value on `,` to create a list of checksums.
    8. This list of keys/values is the qualifiers.
5. Split the remainder once from left on `:`; the right side is the remainder.
6. The left side lowercased is the scheme. It should be exactly "`pkg:`".
7. Strip the remainder from leading and trailing `/`.
8. Split this once from left on `/`; the right side is the remainder.
9. The left side lowercased is the type.
10. Split the remainder once from right on `@`, if present; the left side is the remainder.
11. If the right side is not empty, it contains version information:
    1. Percent-decode the string.
    2. UTF-8-decode this.
    3. This is the version.
12. Split the remainder once from right on `/`, if present; the left side is the remainder.
13. The right side contains name information.
14. Percent-decode the name string.
15. UTF-8-decode this.
16. Apply type-specific normalization, if needed.
17. This is the name.

18. If the remainder is not empty, it contains namespace information:

    1. Split the remainder on `/` to a list of segments.
    2. Discard any empty segment.
    3. Percent-decode each segment.
    4. UTF-8-decode each of these.
    5. Apply type-specific normalization to each segment, if needed.
    6. Join segments with `/`.
    7. This is the namespace.

E.8 Examples

The following list includes some valid *purl* examples:

- `pkg:bitbucket/birkenfeld/pygments-main@244fd47e07d1014f0aed9c`
- `pkg:deb/debian/curl@7.50.3-1?arch=i386&distro=jessie`
- `pkg:gem/ruby-advisory-db-check@0.12.4`
- `pkg:github/package-url/purl-spec@244fd47e07d1004f0aed9c`
- `pkg:golang/google.golang.org/genproto#googleapis/api/annotations`
- `pkg:maven/org.apache.xmlgraphics/batik-anim@1.9.1?packaging=sources`
- `pkg:npm/foobar@12.3.1`
- `pkg:nuget/EnterpriseLibrary.Common@6.0.1304`
- `pkg:pypi/django@1.11.1`
- `pkg:rpm/fedora/curl@7.50.3-1.fc25?arch=i386&distro=fedora-25`

E.9 Original license

This specification is based on the texts published in the https://github.com/package-url/purl-spec online repository. The original license and attribution are reproduced below:

Annex F

History with OMG, Motivation and Rational (Informative)

The OMG and CISQ involvement in developing this specification had its start due to a need that came from the several years of work in the Department of Commerce's National Telecommunications and Information Administration (NTIA) in creating an Initiative to Improve Software Component Transparency in July of 2018 [6]. That effort actually was the culmination of several earlier attempts to get software transparency, updatability and bill of materials as requirements in safety critical sectors like automotive and healthcare as early as 2013/2014 [2, 3] with many talks and papers written for and against them and discussions. With the launch of the NTIA Software Component Transparency Initiative there was a major increase in the energy and coordination of those proposing software bill of materials (SBOM) as a key element of communication across the different participants in software supply chains. These meetings, which started with a public meeting in Washington DC consisted primarily of vendors of software and customers of those vendors. It was this mix of participants that struck us that these efforts were missing an important community member if they were to make SBOMs successful and useful – they seemed to be missing the organizations who create the tools for developing software.

To address this gap, over the winter and spring of 2019, we crafted a market analysis of the software development tooling ecosystem and documented usage scenarios to drive the functionality needed for an SBOM standard usable by tools to talk to other tools and bring speed and agility into the discussion of software transparency and assurance about the information itself. This information was used to present to the Systems Assurance Platform Task Force (PTF) and the Architecture Driven Modernization PTF in March and June of 2019. The paper "Standardizing SBOM within the SW Development Tooling Ecosystem", which captured this work, was later published by MITRE [1] and included 8 core usage scenarios for SBOMs as well as a discussion of the various roles were in the software creation tooling ecosystem. This paper and its various pre-publication drafts were used as a discussion starter to garner interest and participation in the Tool-to-Tool (3T) Software Bill of Materials Exchange effort [4]. The 3T-SBOM Exchange effort was co-sponsored by CISQ and OMG and launched in the fall of 2019 with three to four weekly meetings working the various facets of SBOMs. Over the next two years the 3T-SBOM community, which included over 30 organizations that develop and integrate software creation tooling and infrastructure, developed a 3T-SBOM core model (shown in Figure 13) in September of 2020 that had seven basic concepts connected together to address the usage scenarios outlined for the project.

While the 3T-SBOM community was working to develop their model, the work within the NTIA Software Component Transparency effort also met in numerous weekly virtual meetings to discuss the various aspects of SBOMs, their use, the roles of different players in the lifecycle of an SBOM and the need to educate the world about SBOMs. This was captured in the NTIA Software Bill Of Materials web page. [7]

In late 2020 and much of 2021 the world of software security turned its attention to the software supply chain attack on the Solar Winds Corporation [5] and the need to prevent similar types of attacks in the future. The United States Government responded to this and other similar attacks by issuing Executive Order 14028 in May 2021 [12] calling for stronger software security practices for products used by the government and that the software have SBOMs with them. The Executive Order required that "Within 60 days of the date of this order, the Secretary of Commerce, in coordination with the Assistant Secretary for Communications and Information and the Administrator of the National Telecommunications and Information Administration, shall publish minimum elements for an SBOM." This was done leveraging the community work that NTIA had been doing with industry for the past 34 months and set the new requirements for SBOM capabilities. [11]

Over the 2019-2020 timespan, some of the organization in the 3T-SBOM community were also working within the Linux Foundation's Software Package Data Exchange (SPDX®) open-source effort to evolve their previous work. Started in 2010 to help organizations developing software that planned to incorporate open source software make sure that the licenses for that open source software were appropriate for how the organizations planned to use them in their own offerings, the SPDX community developed a series of software products, specifications, and capabilities to address this area. The first published work was a version 1.0 specification in August of 2011; followed by 1.1 version a year later; a 1.2 version in October 2013; a 2.0 version in 2015; and 2.1 version in 2016. The 2.2 version of the specification was published in 2020 to address the required SBOM minimum elements. The 2.2.1 version of SPDX specification was published through the Linux Foundation's new Joint Development Foundation and sent to ISO under the Publicly Available Standard (PAS) process with it eventually being republished as "ISO/IEC 5962:2021 - Information technology — SPDX®" in 2021.

Through the common members in 3T-SBOM and the Linux Foundation's SPDX effort many of the concepts around SBOMs flowed back and forth between the two resulting in a draft core model for SPDX 3.0 in September of 2020 that had the same seven basic concepts connected together that were in the 3T-SBOM core model. Figure 14 shows the state of the SPDX 3.0 core model at that time.

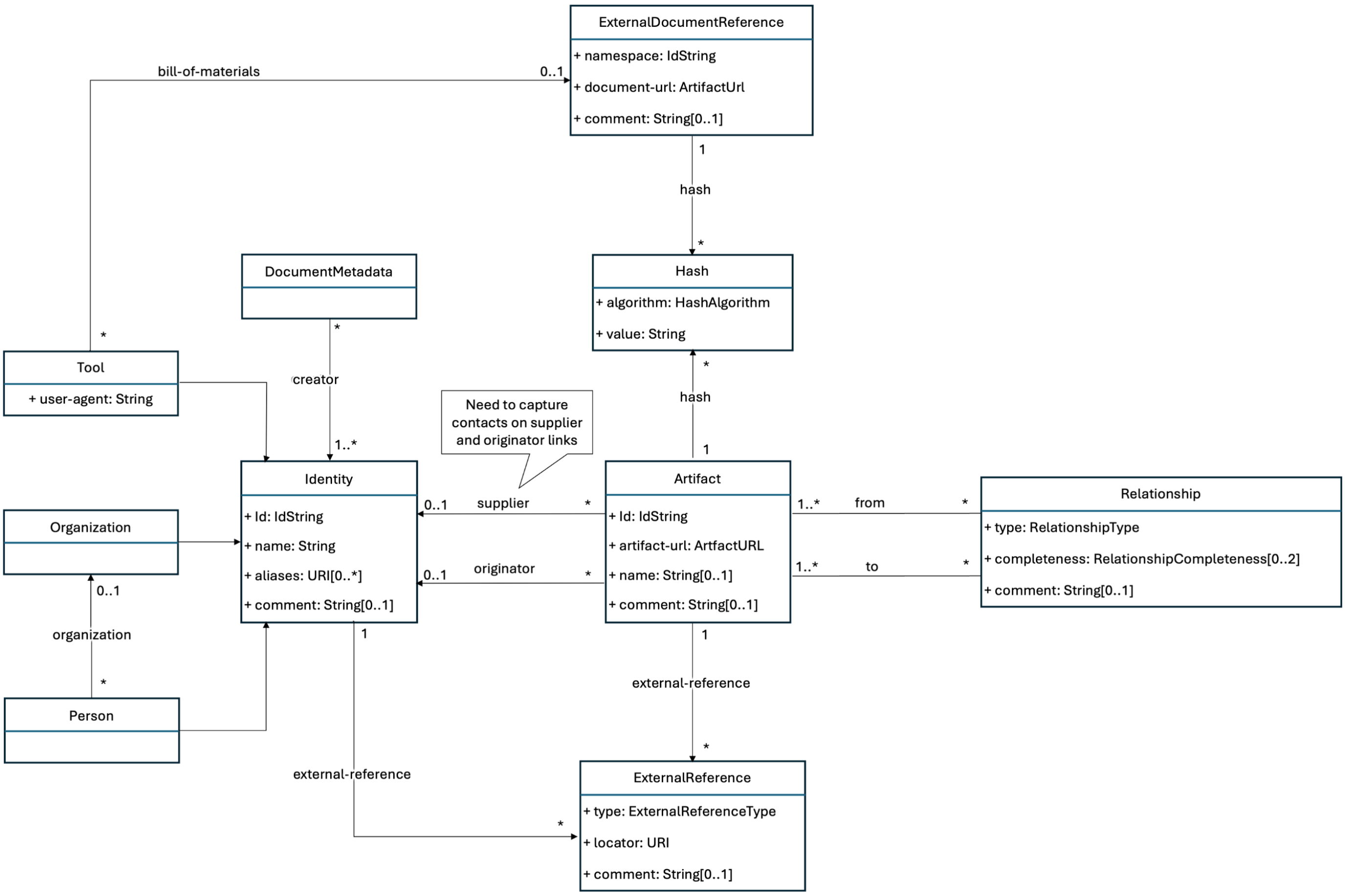

Figure F.1: 3T-SBOM draft core model (circa Sep 2020)

The similarities and alignment of the two group's work (shown in Figure 15 below) was brought to the attention of both teams and after long discussions about each other's efforts, goals, and approach to creating a standard for today, both agreed in principle to join together under the SPDX 3.0 label but to make several changes in the way the SPDX community activities were run as well as how the resulting specification would be vetted.

Specifically, the SPDX community revised their charter to align with the processes of a Standards Development Organization, electing new chairs and adding the OMG Architecture Board review as a gating factor in the publication of SPDX 3.0 and subsequent versions.

The merged activities of the two group slid together the beginning weeks of 2021 with activities generally moving forward but occasionally stalling while the larger group worked through issues that one or the other hadn't discussed or had a different opinion about. Eventually, after releasing SPDX 2.3 in August of 2022 with updates that brought some of the concepts and capabilities slated for SPDX 3.0 to the community in preparation of the shift that SPDX 3.0 represents, the first release candidate of SPDX 3.0 was released in May of 2023. Within the SPDX community, which is both a standards creation organization as well as a community of open source developers, a release candidate offers an opportunity for implementors of SPDX, both new and old, to review the work and determine whether there were parts that were unclear or that would be extremely burdensome to implement.

Based on the comments and change requests from the initial candidate release several areas of the model were revised and reworked, resulting in a release candidate 2 of SPDX 3.0 in February of 2024. This release candidate will give tool creators and those who maintain the support libraries for working with SPDX time to start revising their projects in advance of the final version of the specification. For those not following the inner workings, debates, and discussion of the combined 3T-SBOM and SPDX 3.0 working group for the last 3 years there will be a dramatic change in the SPDX model as it goes from SPDX 2.3 to SPDX 3.0, as shown by looking at Figure 16's left-side (SPDX 2.3 model) compared to its right-side (SPDX 3.0), shifting the SPDX name from Software Package Data Exchange to System Package Data Exchange and the scope of items it can convey in a bill of materials.

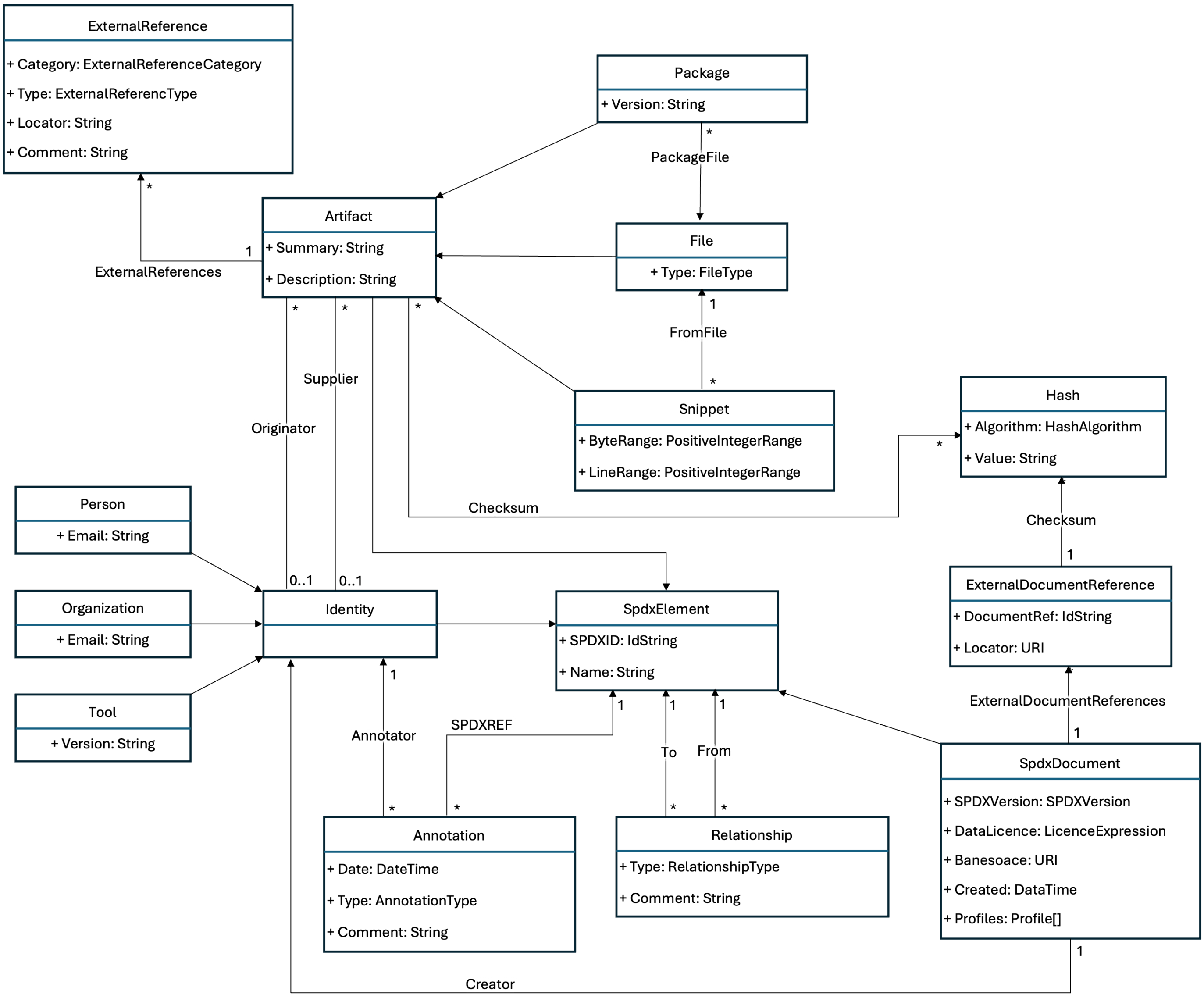

Figure F.2: SPDX 3.0 draft core model (circa Sep 2020)

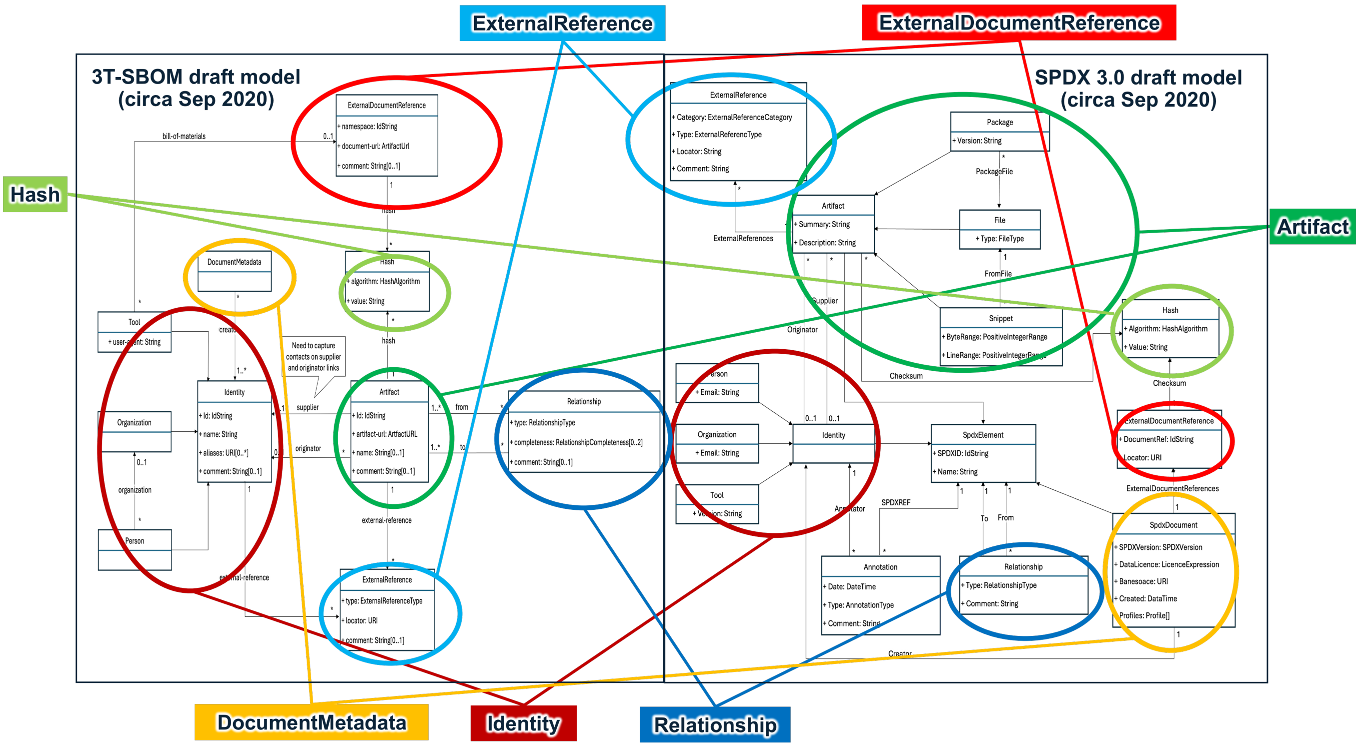

Figure F.3: Correspondence between 3T-SBOM and SPDX 3.0 draft models (circa Sep 2020)

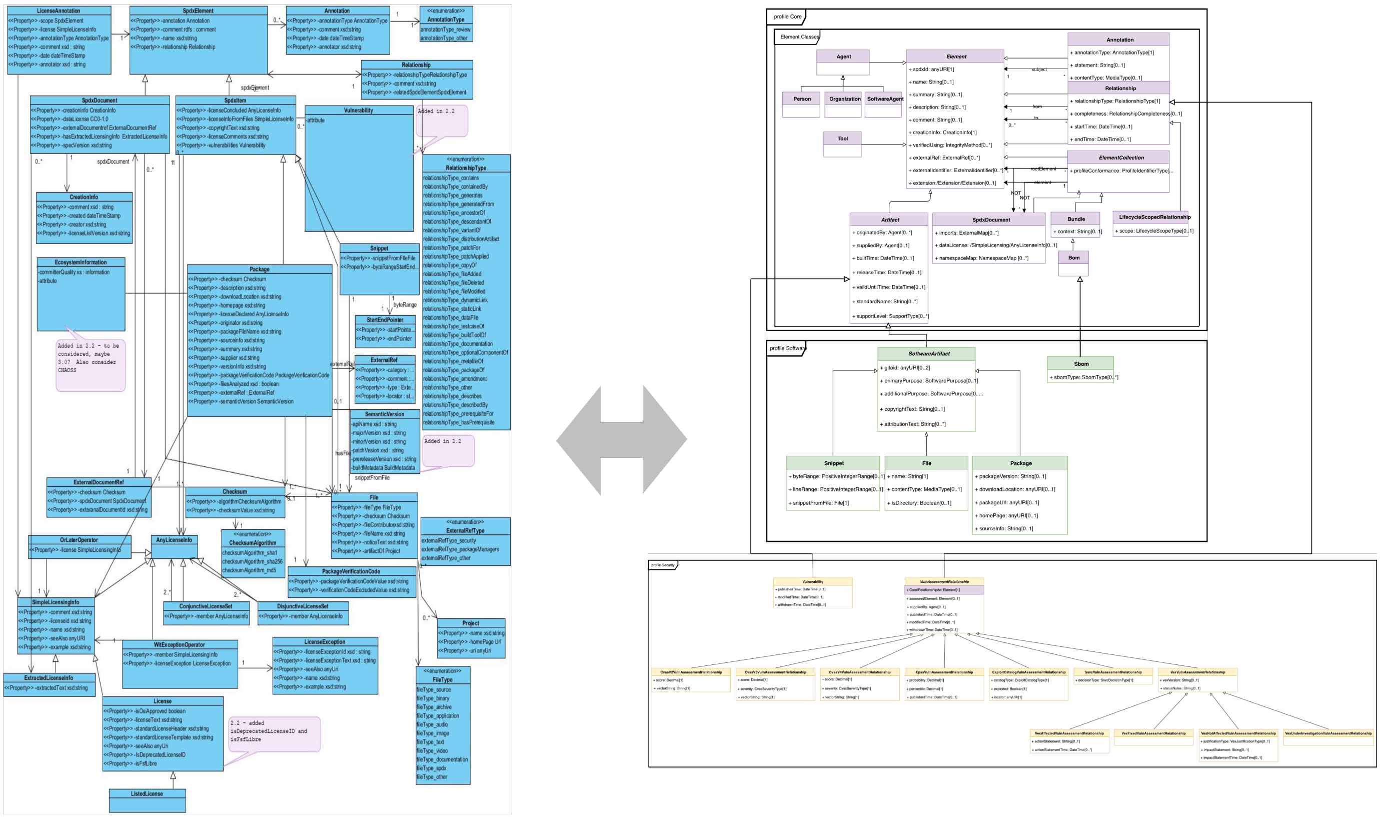

Figure F.4: SPDX 2.3 Model compared to the SPDX 3.0 Model

Annex G

Community Specification License 1.0

The Purpose of this License. This License sets forth the terms under which 1) Contributor will participate in and contribute to the development of specifications, standards, best practices, guidelines, and other similar materials under this Working Group, and 2) how the materials developed under this License may be used. It is not intended for source code. Capitalized terms are defined in the License's last section.

1. Copyright.

1.1. Copyright License. Contributor grants everyone a non-sublicensable, perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as expressly stated in this License) copyright license, without any obligation for accounting, to reproduce, prepare derivative works of, publicly display, publicly perform, and distribute any materials it submits to the full extent of its copyright interest in those materials. Contributor also acknowledges that the Working Group may exercise copyright rights in the Specification, including the rights to submit the Specification to another standards organization.

1.2. Copyright Attribution. As a condition, anyone exercising this copyright license must include attribution to the Working Group in any derivative work based on materials developed by the Working Group. That attribution must include, at minimum, the material's name, version number, and source from where the materials were retrieved. Attribution is not required for implementations of the Specification.

2. Patents.

2.1. Patent License.

2.1.1. As a Result of Contributions.

2.1.1.1. As a Result of Contributions to Draft Specifications. Contributor grants Licensee a non-sublicensable, perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as expressly stated in this License) license to its Necessary Claims in 1) Contributor's Contributions and 2) to the Draft Specification that is within Scope as of the date of that Contribution, in both cases for Licensee's Implementation of the Draft Specification, except for those patent claims excluded by Contributor under Section 3.

2.1.1.2. For Approved Specifications. Contributor grants Licensee a non-sublicensable, perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as expressly stated in this License) license to its Necessary Claims included the Approved Specification that are within Scope for Licensee's Implementation of the Approved Specification, except for those patent claims excluded by Contributor under Section 3.

2.1.2. Patent Grant from Licensee. Licensee grants each other Licensee a non-sublicensable, perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as expressly stated in this License) license to its Necessary Claims for its Implementation, except for those patent claims excluded under Section 3.

2.1.3. Licensee Acceptance. The patent grants set forth in Section 2.1 extend only to Licensees that have indicated their agreement to this License as follows:

2.1.3.1. Source Code Distributions. For distribution in source code, by including this License in the root directory of the source code with the Implementation;

2.1.3.2. Non-Source Code Distributions. For distribution in any form other than source code, by including this License in the documentation, legal notices, via notice in the software, and/or other written materials provided with the Implementation; or

2.1.3.3. Via Notices.md. By issuing pull request or commit to the Specification's repository's Notices.md file by the Implementer's authorized representative, including the Implementer's name, authorized individual and system identifier, and Specification version.

2.1.4. Defensive Termination. If any Licensee files or maintains a claim in a court asserting that a Necessary Claim is infringed by an Implementation, any licenses granted under this License to the Licensee are immediately terminated unless 1) that claim is directly in response to a claim against Licensee regarding an Implementation, or 2) that claim was brought to enforce the terms of this License, including intervention in a third-party action by a Licensee.

2.1.5. Additional Conditions. This License is not an assurance (i) that any of Contributor's copyrights or issued patent claims cover an Implementation of the Specification or are enforceable or (ii) that an Implementation of the Specification would not infringe intellectual property rights of any third party.

2.2. Patent Licensing Commitment. In addition to the rights granted in Section 2.1, Contributor agrees to grant everyone a no charge, royalty-free license on reasonable and non-discriminatory terms to Contributor's Necessary Claims that are within Scope for: 1) Implementations of a Draft Specification, where such license applies only to those Necessary Claims infringed by implementing Contributor's Contribution(s) included in that Draft Specification, and 2) Implementations of the Approved Specification.

This patent licensing commitment does not apply to those claims subject to Contributor's Exclusion Notice under Section 3.

2.3. Effect of Withdrawal. Contributor may withdraw from the Working Group by issuing a pull request or commit providing notice of withdrawal to the Working Group repository's Notices.md file. All of Contributor's existing commitments and obligations with respect to the Working Group up to the date of that withdrawal notice will remain in effect, but no new obligations will be incurred.

2.4. Binding Encumbrance. This License is binding on any future owner, assignee, or party who has been given the right to enforce any Necessary Claims against third parties.

3. Patent Exclusion.

3.1. As a Result of Contributions. Contributor may exclude Necessary Claims from its licensing commitments incurred under Section 2.1.1 by issuing an Exclusion Notice within 45 days of the date of that Contribution. Contributor may not issue an Exclusion Notice for any material that has been included in a Draft Deliverable for more than 45 days prior to the date of that Contribution.

3.2. As a Result of a Draft Specification Becoming an Approved Specification. Prior to the adoption of a Draft Specification as an Approved Specification, Contributor may exclude Necessary Claims from its licensing commitments under this Agreement by issuing an Exclusion Notice. Contributor may not issue an Exclusion Notice for patents that were eligible to have been excluded pursuant to Section 3.1.

4. Source Code License. Any source code developed by the Working Group is solely subject the source code license included in the Working Group's repository for that code. If no source code license is included, the source code will be subject to the MIT License.

5. No Other Rights. Except as specifically set forth in this License, no other express or implied patent, trademark, copyright, or other rights are granted under this License, including by implication, waiver, or estoppel.

6. Antitrust Compliance. Contributor acknowledge that it may compete with other participants in various lines of business and that it is therefore imperative that they and their respective representatives act in a manner that does not violate any applicable antitrust laws and regulations. This License does not restrict any Contributor from engaging in similar specification development projects. Each Contributor may design, develop, manufacture, acquire or market competitive deliverables, products, and services, and conduct its business, in whatever way it chooses. No Contributor is obligated to announce or market any products or services. Without limiting the generality of the foregoing, the Contributors agree not to have any discussion relating to any product pricing, methods or channels of product distribution, division of markets, allocation of customers or any other topic that should not be discussed among competitors under the auspices of the Working Group.

7. Non-Circumvention. Contributor agrees that it will not intentionally take or willfully assist any third party to take any action for the purpose of circumventing any obligations under this License.

8. Representations, Warranties and Disclaimers.

8.1. Representations, Warranties and Disclaimers. Contributor and Licensee represents and warrants that 1) it is legally entitled to grant the rights set forth in this License and 2) it will not intentionally include any third party materials in any Contribution unless those materials are available under terms that do not conflict with this License. IN ALL OTHER RESPECTS ITS CONTRIBUTIONS ARE PROVIDED "AS IS." The entire risk as to implementing or otherwise using the Contribution or the Specification is assumed by the implementer and user. Except as stated herein, CONTRIBUTOR AND LICENSEE EXPRESSLY DISCLAIM ANY WARRANTIES (EXPRESS, IMPLIED, OR OTHERWISE), INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, CONDITIONS OF QUALITY, OR TITLE, RELATED TO THE CONTRIBUTION OR THE SPECIFICATION. IN NO EVENT WILL ANY PARTY BE LIABLE TO ANY OTHER PARTY FOR LOST PROFITS OR ANY FORM OF INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER FROM ANY CAUSES OF ACTION OF ANY KIND WITH RESPECT TO THIS AGREEMENT, WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, AND WHETHER OR NOT THE OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. Any obligations regarding the transfer, successors in interest, or assignment of Necessary Claims will be satisfied if Contributor or Licensee notifies the transferee or assignee of any patent that it knows contains Necessary Claims or necessary claims under this License. Nothing in this License requires Contributor to undertake a patent search. If Contributor is 1) employed by or acting on behalf of an employer, 2) is making a Contribution under the direction or control of a third party, or 3) is making the Contribution as a consultant, contractor, or under another similar relationship with a third party, Contributor represents that they have been authorized by that party to enter into this License on its behalf.

8.2. Distribution Disclaimer. Any distributions of technical information to third parties must include a notice materially similar to the following: "THESE MATERIALS ARE PROVIDED "AS IS." The Contributors and Licensees expressly disclaim any warranties (express, implied, or otherwise), including implied warranties of merchantability, non-infringement, fitness for a particular purpose, or title, related to the materials. The entire risk as to implementing or otherwise using the materials is assumed by the implementer and user. IN NO EVENT WILL THE CONTRIBUTORS OR LICENSEES BE LIABLE TO ANY OTHER PARTY FOR LOST PROFITS OR ANY FORM OF INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER FROM ANY CAUSES OF ACTION OF ANY KIND WITH RESPECT TO THIS DELIVERABLE OR ITS GOVERNING AGREEMENT, WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, AND WHETHER OR NOT THE OTHER MEMBER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE."

9. Definitions.

9.1. Affiliate. "Affiliate" means an entity that directly or indirectly Controls, is Controlled by, or is under common Control of that party.

9.2. Approved Specification. "Approved Specification" means the final version and contents of any Draft Specification designated as an Approved Specification as set forth in the accompanying Governance.md file.

9.3. Contribution. "Contribution" means any original work of authorship, including any modifications or additions to an existing work, that Contributor submits for inclusion in a Draft Specification, which is included in a Draft Specification or Approved Specification.

9.4. Contributor. "Contributor" means any person or entity that has indicated its acceptance of the License 1) by making a Contribution to the Specification, or 2) by entering into the Community Specification Contributor License Agreement for the Specification. Contributor includes its Affiliates, assigns, agents, and successors in interest.

9.5. Control. "Control" means direct or indirect control of more than 50% of the voting power to elect directors of that corporation, or for any other entity, the power to direct management of such entity.

9.6. Draft Specification. "Draft Specification" means all versions of the material (except an Approved Specification) developed by this Working Group for the purpose of creating, commenting on, revising, updating, modifying, or adding to any document that is to be considered for inclusion in the Approved Specification.

9.7. Exclusion Notice. "Exclusion Notice" means a written notice made by making a pull request or commit to the repository's Notices.md file that identifies patents that Contributor is excluding from its patent licensing commitments under this License. The Exclusion Notice for issued patents and published applications must include the Draft Specification's name, patent number(s) or title and application number(s), as the case may be, for each of the issued patent(s) or pending patent application(s) that the Contributor is excluding from the royalty-free licensing commitment set forth in this License. If an issued patent or pending patent application that may contain Necessary Claims is not set forth in the Exclusion Notice, those Necessary Claims shall continue to be subject to the licensing commitments under this License. The Exclusion Notice for unpublished patent applications must provide either: (i) the text of the filed application; or (ii) identification of the specific part(s) of the Draft Specification whose implementation makes the excluded claim a Necessary Claim. If (ii) is chosen, the effect of the exclusion will be limited to the identified part(s) of the Draft Specification.

9.8. Implementation. "Implementation" means making, using, selling, offering for sale, importing or distributing any implementation of the Specification 1) only to the extent it implements the Specification and 2) so long as all required portions of the Specification are implemented.

9.9. License. "License" means this Community Specification License.

9.10. Licensee. "Licensee" means any person or entity that has indicated its acceptance of the License as set forth in Section 2.1.3. Licensee includes its Affiliates, assigns, agents, and successors in interest.

9.11. Necessary Claims. "Necessary Claims" are those patent claims, if any, that a party owns or controls, including those claims later acquired, that are necessary to implement the required portions (including the required elements of optional portions) of the Specification that are described in detail and not merely referenced in the Specification.

9.12. Specification. "Specification" means a Draft Specification or Approved Specification included in the Working Group's repository subject to this License, and the version of the Specification implemented by the Licensee.

9.13. Scope. "Scope" has the meaning as set forth in the accompanying Scope.md file included in this Specification's repository. Changes to Scope do not apply retroactively. If no Scope is provided, each Contributor's Necessary Claims are limited to that Contributor's Contributions.

9.14. Working Group. "Working Group" means this project to develop specifications, standards, best practices, guidelines, and other similar materials under this License.

Annex H

Creative Commons Attribution License 3.0 Unported

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE (“CCPL” OR “LICENSE”). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1) **Definitions**

a. **“Adaptation”** means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image (“synching”) will be considered an Adaptation for the purpose of this License.

b. **“Collection”** means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c. **“Distribute”** means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

d. **“Licensor”** means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

e. **“Original Author”** means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

f. **“Work”** means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. **“You”** means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

h. **“Publicly Perform”** means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. **“Reproduce”** means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2) **Fair Dealing Rights**. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3) **License Grant**. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;

b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked “The original work was translated from English to Spanish,” or a modification could indicate “The original work has been modified.”;

c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,

d. to Distribute and Publicly Perform Adaptations.

e. For the avoidance of doubt:

f. **Non-waivable Compulsory License Schemes**. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

ii. **Waivable Compulsory License Schemes**. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

iii. **Voluntary License Schemes**. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4) **Restrictions**. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.

b. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv) , consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4 (b) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

c. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5) **Representations, Warranties and Disclaimer**

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6) **Limitation on Liability**. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7) **Termination**

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8) **Miscellaneous**

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.