Date: June 2024

System Package Data Exchange (SPDX®),

V3.0 – beta 1

OMG Document Number: ptc/24-06-08

Normative Reference: https://www.omg.org/spec/SPDX/

Associated Normative Machine Consumable Files:

https://www.omg.org/spec/SPDX/202403xx/SPDX_ontology_rdf.ttl

This OMG document replaces the submission document (admtf/24-03-15). It is an OMG Adopted Beta Specification and is currently in the finalization phase. Comments on the content of this document are welcome and should be directed to issues@omg.org by July 26th, 2024.

You may view the pending issues for this specification from the OMG revision issues web page https://issues.omg.org/issues/lists.

The FTF Recommendation and Report for this specification will be published in November 2024. If you are reading this after that date, please download the available specification from the OMG Specifications Catalog.

USE OF SPECIFICATION - TERMS, CONDITIONS & NOTICES

LICENSES

PATENTS

GENERAL USE RESTRICTIONS

DISCLAIMER OF WARRANTY

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE OBJECT MANAGEMENT GROUP AND THE COMPANIES LISTED ABOVE MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE. IN NO EVENT SHALL THE OBJECT MANAGEMENT GROUP OR ANY OF THE COMPANIES LISTED ABOVE BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The entire risk as to the quality and performance of software developed using this specification is borne by you. This disclaimer of warranty constitutes an essential part of the license granted to you to use this specification.

RESTRICTED RIGHTS LEGEND

Use, duplication or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c) (1) (ii) of The Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 or in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clauses at 48 C.F.R. 52.227-19 or as specified in 48 C.F.R. 227- 7202-2 of the DoD F.A.R. Supplement and its successors, or as specified in 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors, as applicable. The specification copyright owners are as indicated above and may be contacted through the Object Management Group, 9C Medway Road, PMB 274, Milford, MA 01757, U.S.A.

TRADEMARKS

CORBA®, CORBA logos®, FIBO®, Financial Industry Business Ontology®, FINANCIAL INSTRUMENT GLOBAL IDENTIFIER®, IIOP®, IMM®, Model Driven Architecture®, MDA®, Object Management Group®, OMG®, OMG Logo®, SoaML®, SOAML®, SysML®, UAF®, Unified Modeling Language®, UML®, UML Cube Logo®, VSIPL®, and XMI® are registered trademarks of the Object Management Group, Inc. SPDX® is a registered trademark of the Linux Foundation.

For a complete list of trademarks, see: https://www.omg.org/legal/tm_list.htm. All other products or company names mentioned are used for identification purposes only, and may be trademarks of their respective owners.

COMPLIANCE

The copyright holders listed above acknowledge that the Object Management Group (acting itself or through its designees) is and shall at all times be the sole entity that may authorize developers, suppliers and sellers of computer software to use certification marks, trademarks or other special designations to indicate compliance with these materials.

Software developed under the terms of this license may claim compliance or conformance with this specification if and only if the software compliance is of a nature fully matching the applicable compliance points as stated in the specification. Software developed only partially matching the applicable compliance points may claim only that the software was based on this specification, but may not claim compliance or conformance with this specification. In the event that testing suites are implemented or approved by Object

Management Group, Inc., software developed using this specification may claim compliance or conformance with the specification only if the software satisfactorily completes the testing suites.

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.

Table of Contents

Table of Figures

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

1 Scope

This Software Package Data Exchange® (SPDX®) specification defines a standard data format for communicating the component and metadata information associated with software packages. An SPDX document can be associated with a set of software packages, files or snippets and contains information about the software in the SPDX format described in this specification.

1.1 General

Companies and organizations (collectively "Organizations") are widely using and reusing open source and other software packages. Accurate identification of software is key for many supply chain processes. Vulnerability remediation starts with knowing the details of which version of software is in use on a system. Compliance with the associated licenses requires a set of analysis activities and due diligence that each Organization performs independently, which may include a manual and/or automated scan of software and identification of associated licenses followed by manual verification. Software development teams across the globe use the same open source packages, but little infrastructure exists to facilitate collaboration on the analysis or share the results of these analysis activities. As a result, many groups are performing the same work leading to duplicated efforts and redundant information. With this document, the SPDX workgroup, a combined effort of the Linux Foundation SPDX group and the OMG/CISQ Tool to Tool effort, has created a data exchange format so that information about software packages and related content may be collected and shared in a common format with the goal of saving time and improving data accuracy.

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 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, 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 from software to many additional aspects like data sets, AI models, security, licencing, and build informaton.

2 Conformance

2.1 Introduction

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

- Core and Software Profile (Clauses 7 & 8)
- Security Profile (Clause 9)
- Licencing Profile (Clause 10)
- Dataset Profile (Clause 11)
- AI Profile (Clause 12)
- Build Profile (Clause 13)
- Lite Profile (Clause 14)
- Extension Profile (Clause 15)

The Core and Software Profile are mandatory. All others are optional.

2.2 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.

2.3 Software Profile compliance point

The Software profile includes the definitions of classes, properties and vocabularies for refering to and conveying information about software and is usable by all SPDX profiles when producing or consuming SPDX conten
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 security profile compliance point does not entail support for the Licencing, Data Set, 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.

2.4 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 Licencing, Data Set, AI, Build, Lite, or Extension profiles of the SPDX.

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

2.5 Licencing 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 Licencing 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.

Conformance to the Licencing profile compliance point does not entail support for the Software, Security, Data Set, AI, Build, Lite, or Extension profiles of the SPDX.

This compliance point facilitates interchange of the licencing 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.

2.6 Data Set Profile compliance point

The data set 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 data set 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 data set profile can covey a description or summary of a dataset, including metadata, characteristics, and statistical information about the data. The data set 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 data set profile compliance point does not entail support for the Software, Licencing, Security, AI, Build, Lite, or Extension profiles of the SPDX.

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

2.7 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, Licencing, Security, Data Set, Build, Lite, or Extension profiles of the SPDX.

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

2.8 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 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 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, Licencing, Securotu. Data Set, AI, Lite, or Extension profiles of the SPDX.

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

2.9 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 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 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, Licencing, Security, Data Set, AI, Build, or Extension profiles of the SPDX.

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

2.10 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 Licencing, Security, Data Set, 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.

3 References

3.1 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, The MITRE Corporation, https://cpe.mitre.org/files/cpe-specification_2.2.pdf

NISTIR 7695, Common Platform Enumeration: Naming Specification Version 2.3, NIST, https://csrc.nist.gov/publications/detail/nistir/7695/final

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

NuGet documentation, Microsoft, https://docs.microsoft.com/en-us/nuget/

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

purl (package URL), https://github.com/package-url/purl-spec

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

RFC-1321, The MD5 Message-Digest Algorithm, The Internet Society Network Working Group, https://tools.ietf.org/html/rfc1321

RFC-3174, US Secure Hash Algorithm 1 (SHA1), The Internet Society Network Working Group, https://tools.ietf.org/html/rfc3174

RFC-3986, Uniform Resource Identifier (URI): Generic Syntax, The Internet Society Network Working Group, https://tools.ietf.org/html/rfc3986

RFC-5234, Augmented BNF for Syntax Specifications: ABNF, The Internet Society Network Working Group, https://tools.ietf.org/html/rfc5234

RFC-6234, US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF), The Internet Society Network Working Group, https://tools.ietf.org/html/rfc6234

SoftWare Heritage persistent IDentifiers (SWHIDs), https://docs.softwareheritage.org/devel/swh-model/persistent-identifiers.html

SPDX and RDF Ontology, http://spdx.org/rdf/ontology/spdx-2-3

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

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

3.2 Non-normative References

Software Package Data Exchange (SPDX®) Specification Version 1.0 and 1.1, 1.2, 2.0, 2.1, and 2.2; SPDX.dev, https://spdx.dev/specifications

Open Source Initiative (OSI); https://opensource.org/licenses

4 Terms and Definitions

For the purposes of this specification, the following terms and definitions apply.

4.1 annotations information section

section (4.9) type, an instance of which contains comments about an SPDX document, SPDX file, SPDX package, or SPDX snippet

4.2 field

a piece of information contained in a section (4.9)

4.3 file information section

section (4.9) type, an instance of which contains facts specific to files

4.4 other licensing information detected section

section (4.9) type, an instance of which contains a way to capture information about and refer to licenses that are not on the SPDX license List

4.5 package

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

4.6 package information section

section (4.9) type, an instance of which contains facts that are common properties of a package

4.7 relationships between SPDX elements information section

section (4.9) type, an instance of which contains information on how documents, packages (3.5), files and snippets relate to each other

4.8 review information section

section (4.9) type, an instance of which contains information about persons, organizations or tools that have reviewed a document

4.9 section

a part of this SPDX specification

4.10 snippet information section

section (4.9) type, an instance of which contains facts that are specific to a part of a file

4.11 SPDX document

collection of section (4.8) instances each of which contains information about software organized using the SPDX format (4.11)

4.12 SPDX document creation information section

section (4.9) type, an instance of which contains metadata that associates analysis results with a specific version of an SPDX document (4.11) and license for use, and provides information on how, when, and by whom the SPDX document was created

4.13 SPDX format

the data format defined by this document

4.14 sub-package

a package which is embedded in a larger package

5 Symbols

List of symbols/abbreviations.

3T-SBOM	Tool-to-Tool Software Bill of Material
ABNF	Augmented Backus–Naur form
AI	Artificial Intelligence
BNF	Backus–Naur form
BOM	Bill of Material
CISA	Cybersecurity and Information Security Agency
CISQ	Center for Information and Security Quality
CPE	Common Package Enumeration
CVE	Common Vulnerabilies and Exposures
CVSS	Common Vulnerability Scoring System
EPSS	Exploit Prediction Scoring System
ISO	International Organization for Standardization
JSON-LD	JavaScript Object Notation for Linked Data
KEV	Known Exploited Vulnerabilities
ML	Machine Learning
NISTIR	National Institute of Standards and Technology Internal/Interagency Reports
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 Uniform Resource Identifier
RDF	Resource Description Framework
RFC	Request For Comment
SBOM	Software Bill of Material
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 Heritage persistent IDentifiers

URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VEX	Vulnerability Exploitable eXchange
XML	Extensible Markup Language

6 Additional Information

6.1 Changes to Other OMG Specifications

None.

6.2 Acknowledgments

The following organizations submitted this specification:

- CAST Software
- The MITRE Corporation

The following additional organizations contributed to this specification:

- The Linux Foundation

6.3 Intellectual Property Rights

The SPDX 3.0 specification and related artifacts are available under the OMG's Copyright and Non-Assertion Covenant (see https://www.omg.org/cgi-bin/doc.cgi?ipr for details).

The ontologies themselves are licensed under the Community Specification License 1.0 open-source license agreement, available at https://github.com/spdx/governance/blob/main/1._Community_Specification_License-v1.md.

7 History, Motivation and Rationale

The OMG and CISQ involvement in developing this specification has 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 [1]. That effort actually was the culmination of several earlier attempts to get software transparency, updatability and bill of material 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 Material (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 [4] 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 [5]. 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 1) in September of 2020 that had seven basic concepts connected together to address the usage scenarios outlined for the project.

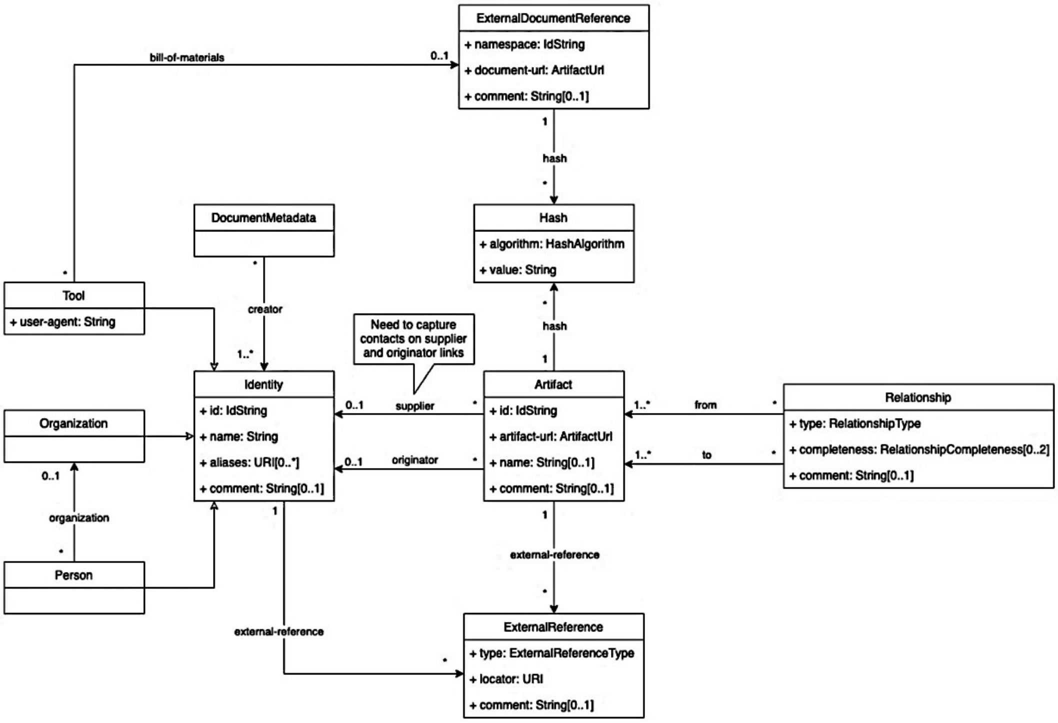


Figure 5 – 3T-SBOM draft core model (circa Sep 2020)

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. [6]

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 [7] 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 [8] 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. [9]

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 2.1 version of SPDX 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 2 shows the state of the SPDX 3.0 core model at that time.

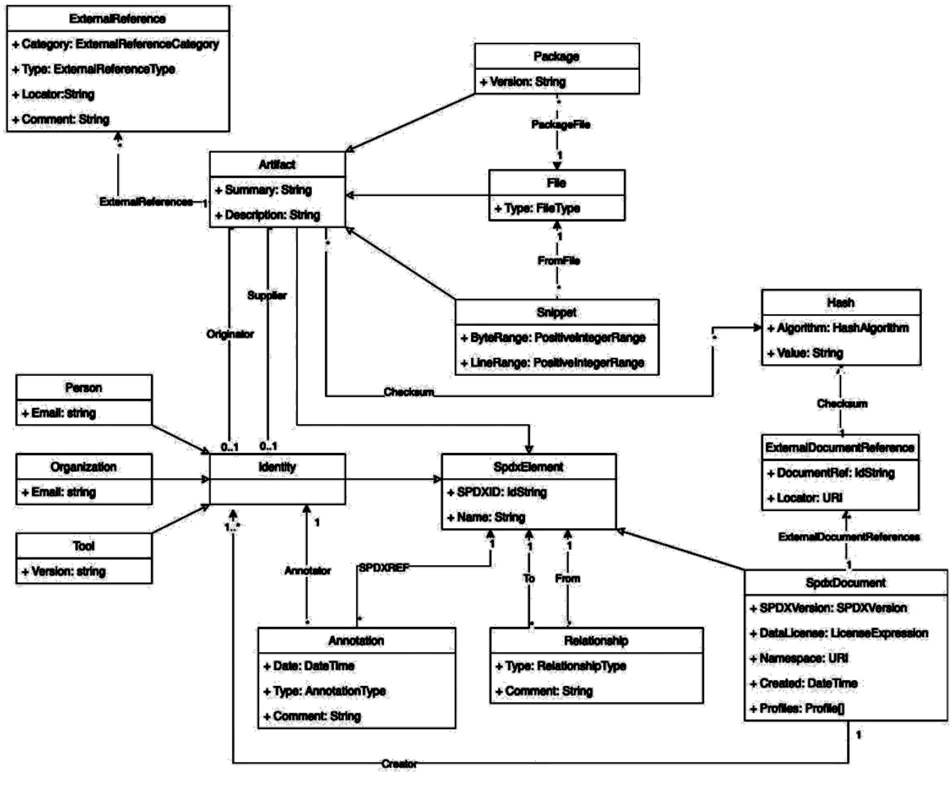


Figure 6 – SPDX 3.0 draft core model (circa Sep 2020)

The similarities and alignment of the two group's work (shown in Figure 3 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.

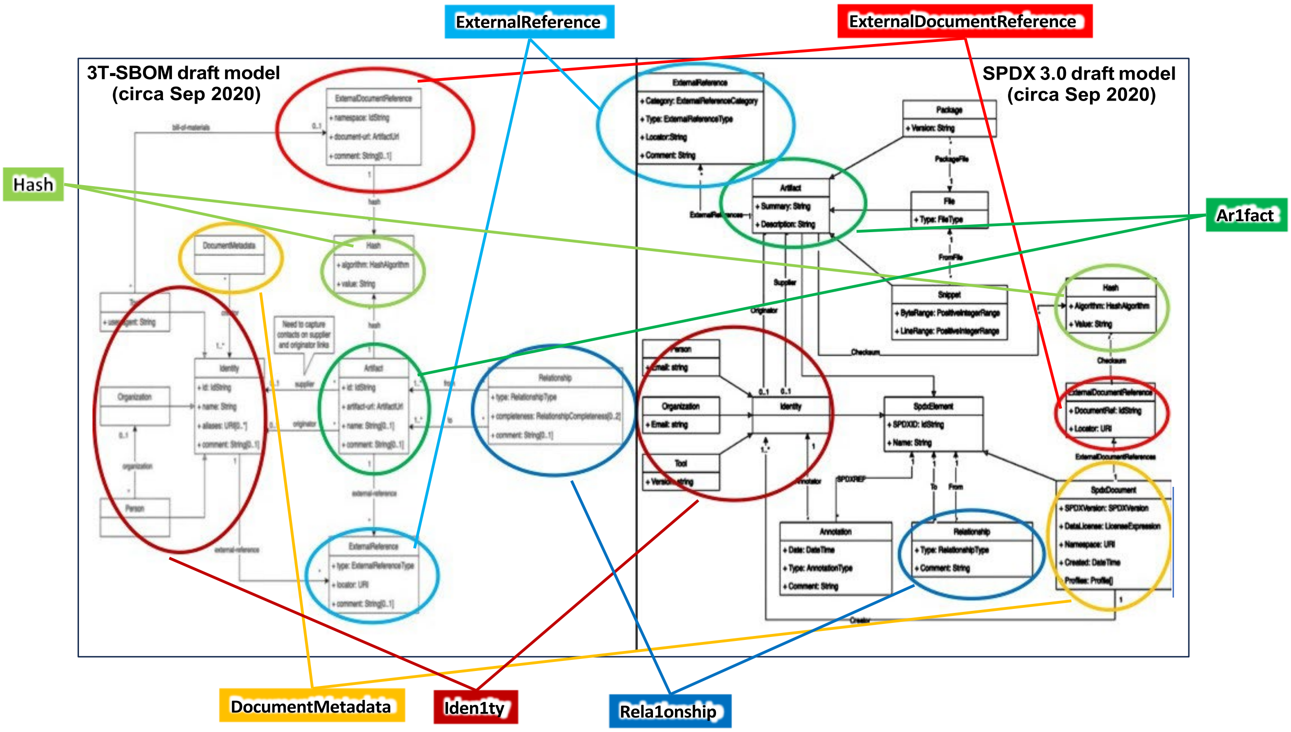


Figure 7 – Correspondence between 3T-SBOM and SPDX 3.0 draft models (circa Sep 2020)

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 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 4'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.

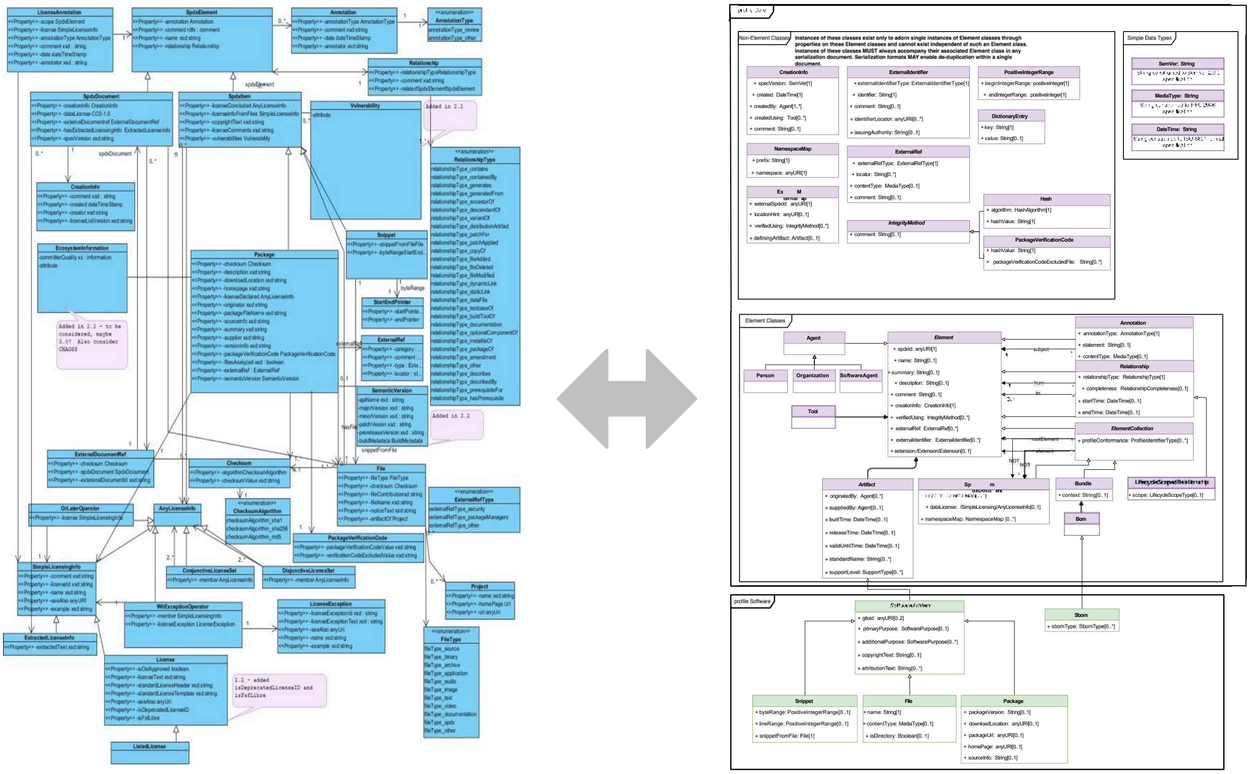


Figure 8 – SPDX 2.3 Model compared to the SPDX 3.0 Model

The SPDX 3.0 model is available at: https://github.com/spdx/spdx-3-model

The SPDX 3.0 ontology is available at: https://github.com/spdx/spdx-spec/tree/development/v3.0/ontology

The SPDX 3.0 specification is available as web pages at: https://spdx.github.io/spdx-spec/v3.0/

8 Core Profile

Summary

The basis for all SPDX profiles.

Description

The Core namespace defines foundational concepts serving as the basis for all SPDX-3.0 profiles. Figure 5 below shows the logical model for Core profile, for the Software profile, and the non-element classes, enumerations, and data types for both.

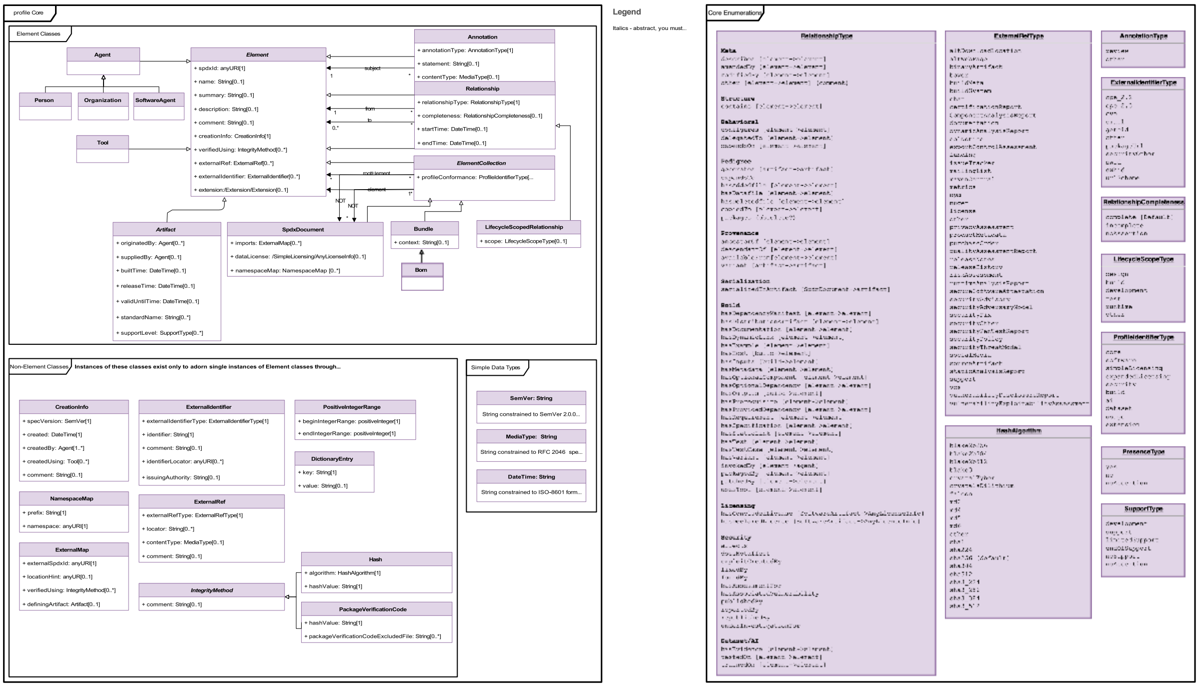

profile Core
Element Classes
Agent
Person
Organization
SoftwareAgent
Tool
Element
+ spdxId: anyURI[1]
+ name: String[0..1]
+ summary: String[0..1]
+ description: String[0..1]
+ comment: String[0..1]
+ creationInfo: CreationInfo[1]
+ verifiedUsing: IntegrityMethod[0..*]
+ externalRef: ExternalRef[0..*]
+ externalIdentifier: ExternalIdentifier[0..*]
+ extension:/Extension/Extension[0..1]
Annotation
+ annotationType: AnnotationType[1]
+ statement: 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
1
*
to
0..*
*
ElementCollection
+ profileConformance: ProfileIdentifierType[...
rootElement
*
*
element
1*
NOT
NOT
*
Artifact
+ originatedBy: Agent[0..*]
+ suppliedBy: Agent[0..1]
+ builtTime: DateTime[0..1]
+ releaseTime: DateTime[0..1]
+ validUntilTime: DateTime[0..1]
+ standardName: String[0..*]
+ supportLevel: SupportType[0..*]
SpdxDocument
+ imports: ExternalMap[0..*]
+ dataLicense: /SimpleLicensing/AnyLicenseInfo[0..1]
+ namespaceMap: NamespaceMap [0..*]
Bundle
+ context: String[0..1]
Bom
LifecycleScopedRelationship
+ scope: LifecycleScopeType[0..1]
Legend
Italics - abstract, you must...
Non-Element Classes
Instances of these classes exist only to adorn single instances of Element classes through...
CreationInfo
+ specVersion: SemVer[1]
+ created: DateTime[1]
+ createdBy: Agent[1..*]
+ createdUsing: Tool[0..*]
+ comment: String[0..1]
ExternalIdentifier
+ externalIdentifierType: ExternalIdentifierType[1]
+ identifier: String[1]
+ comment: String[0..1]
+ identifierLocator: anyURI[0..*]
+ issuingAuthority: String[0..1]
PositiveIntegerRange
+ beginIntegerRange: positiveInteger[1]
+ endIntegerRange: positiveInteger[1]
DictionaryEntry
+ key: String[1]
+ value: String[0..1]
NamespaceMap
+ prefix: String[1]
+ namespace: anyURI[1]
ExternalRef
+ externalRefType: ExternalRefType[1]
+ locator: String[0..*]
+ contentType: MediaType[0..1]
+ comment: String[0..1]
ExternalMap
+ externalSpdxId: anyURI[1]
+ locationHint: anyUR[0..1]
+ verifiedUsing: IntegrityMethod[0..*]
+ definingArtifact: Artifact[0..1]
IntegrityMethod
+ comment: String[0..1]
Hash
+ algorithm: HashAlgorithm[1]
+ hashValue: String[1]
PackageVerificationCode
+ hashValue: String[1]
+ packageVerificationCodeExcludedFile: String[0..*]
Simple Data Types
SemVer: String
String constrained to SemVer 2.0.0...
MediaType: String
String constrained to RFC 2046 spe...
DateTime: String
String constrained to ISO-8601 form...
Core Enumerations
RelationshipType
Meta
describes [element->element]
amendedBy [element->element]
certifiedBy [element->element]
other [element->element] [comment]
Structure
contains [element->element]
Behavioral
configures [element->element]
delegatedTo [element->element]
dependsOn [element->element]
Pedigree
generates [artifact->artifact]
expandedTo
hasAddedFile [element->element]
hasDataFile [element->element]
hasDeletedFile [element->element]
copiedTo [element->element]
packages (obsolete?)
Provenance
ancestorOf [element->element]
descendantOf [element->element]
availableFrom[element->element]
variant [artifact->artifact]
Serialization
serializedInArtifact [SpdxDocument->artifact]
Build
hasDependencyManifest [element->element]
hasDistributionArtifact [element->element]
hasDocumentation [element->element]
hasDynamicLink [element->element]
hasExample [element->element]
hasHost [build->element]
hasInputs [build->element]
hasMetadata [element->element]
hasOptionalComponent [element->element]
hasOptionalDependency [element->element]
hasOutputs [build->element]
hasPrerequisite [element->element]
hasProvidedDependency [element->element]
hasRequirement [element->element]
hasSpecification [element->element]
hasStaticLink [element->element]
hasTest [element->element]
hasTestCase [element->element]
hasVariant [element->element]
invokedBy [element->agent]
packagedBy [element->element]
patchedBy [element->element]
usesTool [element->element]
Licensing
hasConcludedLicense [SoftwareArtifact->AnyLicenseInfo]
hasDeclaredLicense [SoftwareArtifact->AnyLicenseInfo]
Security
affects
doesNotAffect
exploitCreatedBy
fixedBy
foundBy
hasAssessmentFor
hasAssociatedVulnerability
publishedBy
reportedBy
republishedBy
underInvestigationFor
Dataset/AI
hasEvidence [element->element]
testedOn [element->element]
trainedOn [element->element]
ExternalRefType
altDownloadLocation
altWebPage
binaryArtifact
bower
buildMeta
buildSystem
chat
certificationReport
componentAnalysisReport
documentation
dynamicAnalysisReport
eolNotice
exportControlAssessment
funding
issueTracker
mailingList
mavenCentral
metrics
npm
nuget
license
other
privacyAssessment
productMetadata
purchaseOrder
qualityAssessmentReport
releaseNotes
releaseHistory
riskAssessment
runtimeAnalysisReport
secureSoftwareAttestation
securityAdvisory
securityAdversaryModel
securityFix
securityOther
securityPenTestReport
securityPolicy
securityThreatModel
socialMedia
sourceArtifact
staticAnalysisReport
support
vcs
vulnerabilityDisclosureReport
vulnerabilityExploitabilityAssessment
HashAlgorithm
blake2b256
blake2b384
blake2b512
blake3
crystalsKyber
crystalsDilithium
falcon
md2
md4
md5
md6
other
sha1
sha224
sha256 [default]
sha384
sha512
sha3_224
sha3_256
sha3_384
sha3_512
AnnotationType
review
other
ExternalIdentifierType
cpe_2.2
cpe_2.3
cve
email
gitoid
other
packageUrl
securityOther
swhid
swid
urlScheme
RelationshipCompleteness
complete [Default]
incomplete
noAssertion
LifecycleScopeType
design
build
development
test
runtime
other
ProfileIdentifierType
core
software
simpleLicensing
expandedLicensing
security
build
ai
dataset
usage
extension
PresenceType
yes
no
noAssertion
SupportType
development
support
limitedSupport
endOfSupport
noSupport
noAssertion

Figure 9 – Core model profile, non-element classes, enumerations, and single data types

8.1 Core 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/v3/Core/Agent`

Name	Agent
Instantiability	Concrete
SubclassOf	Element

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/v3/Core/Annotation`

Name	Annotation
Instantiability	Concrete
SubclassOf	Element

Properties

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

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/v3/Core/Artifact
```

Name	Artifact
Instantiability	Abstract
SubclassOf	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

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/v3/Core/Bom
```

Name	Bom
Instantiability	Concrete
SubclassOf	Bundle

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/v3/Core/Bundle
```

Name	Bundle
Instantiability	Concrete
SubclassOf	ElementCollection

Properties

Property	Type	minCount	maxCount
context	xsd:string	0	1

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/v3/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

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/v3/Core/DictionaryEntry
```

Name	DictionaryEntry
Instantiability	Concrete

Properties

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/v3/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	*

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 no 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/v3/Core/ElementCollection
```

Name	ElementCollection
Instantiability	Abstract
SubclassOf	Element

Properties

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

8.1.10 ExternalIdentifier

Summary

A reference to a resource 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 uniquely identifies an Element.

Metadata

```
https://spdx.org/rdf/v3/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

8.1.11 ExternalMap

Summary

A map of Element identifiers that are used within a Document but defined external to that Document.

Description

An External Map is a map of Element identifiers that are used within a Document but defined external to that Document. 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/v3/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	*

8.1.12 ExternalRef

Summary

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

Description

An External Reference 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/v3/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	*

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.

Metadata

```
https://spdx.org/rdf/v3/Core/Hash
```

Name	Hash
Instantiability	Concrete
SubclassOf	IntegrityMethod

Properties

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

8.1.14 Integrity Method

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.

Metadata

```
https://spdx.org/rdf/v3/Core/IntegrityMethod
```

Name	IntegrityMethod

Instantiability	Abstract

Properties

Property	Type	minCount	maxCount
comment	xsd:string	0	1

8.1.15 LifecycleScopedRelationship

Summary

Provide context for a relationship that occurs in the software lifecycle.

Description

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

Metadata

```
https://spdx.org/rdf/v3/Core/LifecycleScopedRelationship
```

Name	LifecycleScopedRelationship
Instantiability	Concrete
SubclassOf	Relationship

Properties

Property	Type	minCount	maxCount
Scope	LifecycleScopeType	0	1

8.1.16 NamespaceMap

Summary

A mapping between prefixes and namespace partial URIs.

Description

A namespace map allows the creator of a collection of Elements that could be serialized to suggest a set of shorter identifiers ("prefixes") for particular namespace portions of ElementIDs to be used in SPDX content serialization in order 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 general SPDX serialization guidance at https://spdx.github.io/spdx-3-model/serialization/readme.md and the various serialization format specific .md filed under https://spdx.github.io/spdx-3-model/serialization/(Editors note the URLs will change when the context is publicly available)

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.

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.

8.1.17 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/v3/Core/Organization`

Name	Organization
Instantiability	Concrete
SubclassOf	Agent

8.1.18 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 gitoid 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:

verificationcode = 0 filelist = templist = ""for all files in the package

```
 {

 if file is an "excludes" file, skip it /* exclude SPDX analysis file(s) */append templist
     with "SHA1(file)/n"
```

```
  }
```

sort templist in ascending order by SHA1 value.

8.1.19 Person

Summary

An individual human being.

Description

A Person is an individual human being.

Metadata

`https://spdx.org/rdf/v3/Core/Person`

Name	Person
Instantiability	Concrete
SubclassOf	Agent

8.1.20 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/v3/Core/PositiveIntegerRange`

Name	PositiveIntegerRange
Instantiability	Concrete

Properties

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

8.1.21 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.

Metadata

`https://spdx.org/rdf/v3/Core/Relationship`

Name	Relationship
Instantiability	Concrete
SubclassOf	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	0	*

8.1.22 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/v3/Core/SoftwareAgent`

Name	SoftwareAgent
Instantiability	Concrete
SubclassOf	Agent

8.1.23 Spdx Document

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/v3/Core/SpdxDocument
```

Name	SpdxDocument
Instantiability	Concrete

SubclassOf	ElementCollection

Properties

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

8.1.24 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/v3/Core/Tool
```

Name	Tool
Instantiability	Concrete
SubclassOf	Element

Properties

Property	Type	minCount	maxCount

8.2 Core 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/v3/Core/algorithm
```

Name	algorithm
Nature	ObjectProperty
Range	HashAlgorithm

Referenced

- /Core/Hash

8.2.2 annotationType

Summary

Describes the type of annotation.

Description

An annotationType describes the type of an annotation.

Metadata

```
https://spdx.org/rdf/v3/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/v3/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/v3/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/v3/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/v3/Core/completeness
```

Name	**completeness**
Nature	ObjectProperty
Range	RelationshipCompleteness

Referenced

- /Core/Relationship

8.2.7 contentType

Summary

Specifies the media type of an Element or Property.

Description

ContentType specifies the media type of an Element or Property.

Metadata

```
https://spdx.org/rdf/v3/Core/contentType
```

Name	**contentType**
Nature	DataProperty
Range	MediaType

Referenced

- /Core/Annotation
- /Core/ExternalRef

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/v3/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/v3/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/v3/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/v3/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/v3/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 matt er 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/v3/Core/dataLicense
```

Name	dataLicense

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/v3/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/v3/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/v3/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/v3/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

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

Metadata

```
https://spdx.org/rdf/v3/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

extensionElement specifies an Extension-based characterization of a particular aspect of an

Metadata

```
https://spdx.org/rdf/v3/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/v3/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/v3/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/v3/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/v3/Core/externalRefType
```

Name	**externalRefType**
Nature	ObjectProperty
Range	ExternalRefType

Referenced

- /Core/ExternalRef

8.2.24 externalSpdxId

Summary

Identifies an external Element used within a Document but defined external to that Document.

Description

ExternalSpdxId identifies an external Element used within a Document but defined external to that Document.

Metadata

```
https://spdx.org/rdf/v3/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/v3/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

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

Metadata

```
https://spdx.org/rdf/v3/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/v3/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/v3/Core/identifierLocator
```

Name	**identifierLocator**
Nature	DataProperty
Range	xsd:anyURI

Referenced

- /Core/ExternalIdentifier

8.2.29 imports

Summary

Provides an ExternalMap of Element identifiers.

Description

Imports provides an ExternalMap of Element identifiers that are used within a document but defined external to that document.

Metadata

```
https://spdx.org/rdf/v3/Core/imports
```

Name	**imports**
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/v3/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/v3/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/v3/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/v3/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/v3/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 ElementID.

Description

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

Metadata

```
https://spdx.org/rdf/v3/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/v3/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

OriginatedBy identifies from where or whom the Element originally came.

Metadata

```
https://spdx.org/rdf/v3/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 PackageVerificationCode.

In general, every filename is preceded with a ./, see https://www.ietf.org/rfc/rfc3986.txt for syntax.

Metadata

```
https://spdx.org/rdf/v3/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/v3/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/v3/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/v3/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/v3/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 an SBOM.

Description

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

Metadata

```
https://spdx.org/rdf/v3/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/v3/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

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/v3/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.

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/v3/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 to useful to capture for specific artifacts.

Metadata

```
https://spdx.org/rdf/v3/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 element is applicable /valid.

Metadata

```
https://spdx.org/rdf/v3/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/v3/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/v3/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/v3/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/v3/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/v3/Core/supportLevel
```

Name	**supportLevel**
Nature	DataProperty

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.

Metadata

```
https://spdx.org/rdf/v3/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/v3/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/v3/Core/value
```

Name	**value**
Nature	DataProperty
Range	xsd:string

Referenced

8.2.57 verifiedUsing

Summary

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

Description

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

Metadata

```
https://spdx.org/rdf/v3/Core/verifiedUsing
```

Name	**verifiedUsing**
Nature	ObjectProperty
Range	IntegrityMethod

Referenced

- /Core/Element
- /Core/ExternalMap

8.3 Core 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/v3/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

ExteralIdentifierType specifies the type of an external identifier.

Metadata

```
https://spdx.org/rdf/v3/Core/ExternalIdentifierType
```

Name	ExternalIdentifierType

Entries

- cpe22: https://cpe.mitre.org/files/cpe-specification_2.2.pdf
- cpe23: https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir7695.pdf
- cve: An identifier for a specific software flaw defined within the official CVE Dictionary and that conforms to the CVE specification as defined by https://csrc.nist.gov/glossary/term/cve_id.
- email: https://datatracker.ietf.org/doc/html/rfc3696#section-3
- gitoid: htt ps://www.iana.org/assignments/uri-schemes/prov/gitoid Gitoid stands for Git Object ID and a gitoid of type blob is a unique hash of a binary artifact. A gitoid may represent the software Artifact ID or the OmniBOR Identifier for the software artifact's associated OmniBOR Document; this ambiguity exists because the OmniBOR Document is itself an artifact, and the gitoid of that artifact is its valid identifier. Omnibor is a minimalistic schema to describe software Artifact Dependency Graphs. 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 OmniBOR Document (OmniBOR Identifiers) should be recorded in the SPDX 3.0 Element's ExternalIdentifier property.
- other: Used when the type doesn't match any of the other options.
- packageUrl: https://github.com/package-url/purl-spec

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

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/v3/Core/ExternalRefType
```

Name	ExternalRefType

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.
- 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.
- 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.

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/v3/Core/HashAlgorithm
```

Name	HashAlgorithm

Entries

- blake2b256: blake2b algorithm with a digest size of 256 https://datatracker.ietf.org/doc/html/rfc7693#section-4
- blake2b384: blake2b algorithm with a digest size of 384 https://datatracker.ietf.org/doc/html/rfc7693#section-4
- blake2b512: blake2b algorithm with a digest size of 512 https://datatracker.ietf.org/doc/html/rfc7693#section-4
- blake3: htt ps://github.com/BLAKE3-team/BLAKE3-specs/blob/master/blake3.pdf
- crystalsDilithium: https://pq-crystals.org/dilithium/index.shtml
- crystalsKyber: https://pq-crystals.org/kyber/index.shtml
- falcon: https://falcon-sign.info/falcon.pdf
- md2: https://datatracker.ietf.org/doc/rfc1319/
- md4: https://datatracker.ietf.org/doc/html/rfc1186
- md5: https://datatracker.ietf.org/doc/html/rfc1321
- md6: https://people.csail.mit.edu/rivest/pubs/RABCx08.pdf
- other: any hashing algorithm that does not exist in this list of entries
- sha1: https://datatracker.ietf.org/doc/html/rfc3174

8.3.5 LifecycleScopeType

Summary

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

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 runtime dependency.

Metadata

```
https://spdx.org/rdf/v3/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/v3/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 conformanceProfiles, they are claiming that all contained elements conform to all restrictions defined for that profile.

Metadata

```
https://spdx.org/rdf/v3/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 expanded Licensing profile specification
- extension: the element follows the Extension profile specification
- security: the element follows the Security profile specification
- simpleLicensing: the element follows the simple Licensing profile specification
- software: the element follows the Software profile specification
- usage: the element follows the Usage 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/v3/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 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/v3/Core/RelationshipType
```

Name	RelationshipType

Entries

- affects: (Security/VEX) The from Vulnerability affect each to Element
- amendedBy: The from Element is amended by 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: (Security 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. (e.g. the to invokedBy Relationship is being done on behalf of from)
- 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: (Security/VEX) The from Vulnerability has no impact on each to Element
- expandsTo: The from archive expands out as an artifact described by each to Element
- exploitCreatedBy: (Security) The from Vulnerability has had an exploit created against it by each to Agent
- fixedBy: (Security) Designates a from Vulnerability has been fixed by the to Agent(s)
- fixedIn: (Security/VEX) A from Vulnerability has been fixed in each of the to Element(s)
- foundBy: (Security) 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 is a file added to the from Element (from hasAddedFile to)
- hasAssessmentFor: (Security) Relates a from Vulnerability and each to Element(s) with a security assessment. To be used with VulnAssessmentRelationship types
- hasAssociatedVulnerability: (Security) Used to associate a from Artifact with each to Vulnerability
- hasConcludedLicense: The from Software Artifact 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
- hasDeclaredLicense: The from Software Artifact 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 Element to, (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: (Dataset) 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)
- hasInputs: The from Build has each to Elements 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 hasOptionalComponentto`)
- hasOptionalDependency: The from Element optionally depends on each to Element during a LifecycleScopeType period
- hasOutputs: The from Build element generates each to Element as an output during a LifecycleScopeType period.
- hasPrerequsite: The from Element has a prerequsite on each to Element, during a LifecycleScopeType period
- hasProvidedDependency: The from Element has a dependency on each to Element, but 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 relationhip 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: (Security) Designates a from Vulnerability was made available for public use or reference by each to Agent
- reportedBy: (Security) Designates a from Vulnerability was first reported to a project, vendor, or tracking database for formal identification by each to Agent
- republishedBy: (Security) Designates a from Vulnerability's details were tracked, aggregated, and/or enriched to improve context (i.e. NVD) by a to Agent(s)

- serializedInArtifact: The from SPDXDocument can be found in a serialized form in each to Artifact
- testedOn: (AI, Dataset) The from Element has been tested on the to Element
- trainedOn: (AI, Dataset) The from Element has been trained by the to Element(s)
- underInvestigationFor: (Security/VEX) The from Vulnerability impact is being investigated for each to Element
- 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/v3/Core/SupportType
```

Name	SupportType

Entries

- 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 Core Datatypes

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

Metadata

```
https://spdx.org/rdf/v3/Core/DateTime
```

Name	DateTime
SubclassOf	xsd:string

Format

- Pattern: ^ \ d\ d\ d\ d- \ d\ d- \ d\ d T\ d\ d: \ d\ d: \ d\ d Z$

8.4.2 MediaType

Summary

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

Description

A MediaType is a string constrained to the RFC 2046 specification. It provides a standardized way of indicating the type of content of an Element.

A list of all possible media types is available at https://www.iana.org/assignments/media- types/media-types.xhtml.

Metadata

```
https://spdx.org/rdf/v3/Core/MediaType
```

Name	MediaType
SubclassOf	xsd:string

Format

- Pattern: ^ [^ \ /] +\ /[^ \ /] +$

8.4.3 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.

Metadata

```
https://spdx.org/rdf/v3/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 - z A- Z-] [0 - 9 a - z A- Z-] *) (? : \ . (? : 0 | [1 - 9] \ d*| \ d*[a - z A- Z-] [0 - 9 a - z A- Z-] *)) *)) ? (? : \ +([0 - 9 a - z A- Z-] +(? : \ . [0 - 9 a - z A- Z-] +) *)) ? $

9 Software Profile

Summary

Everything having to do with software.

Description

The Software namespace defines concepts related to software artifacts. Figure 6 below shows the logical model for Core profile, for the Software profile, and the non-element classes, enumerations, and data types for both.

Metadata

Name	Software

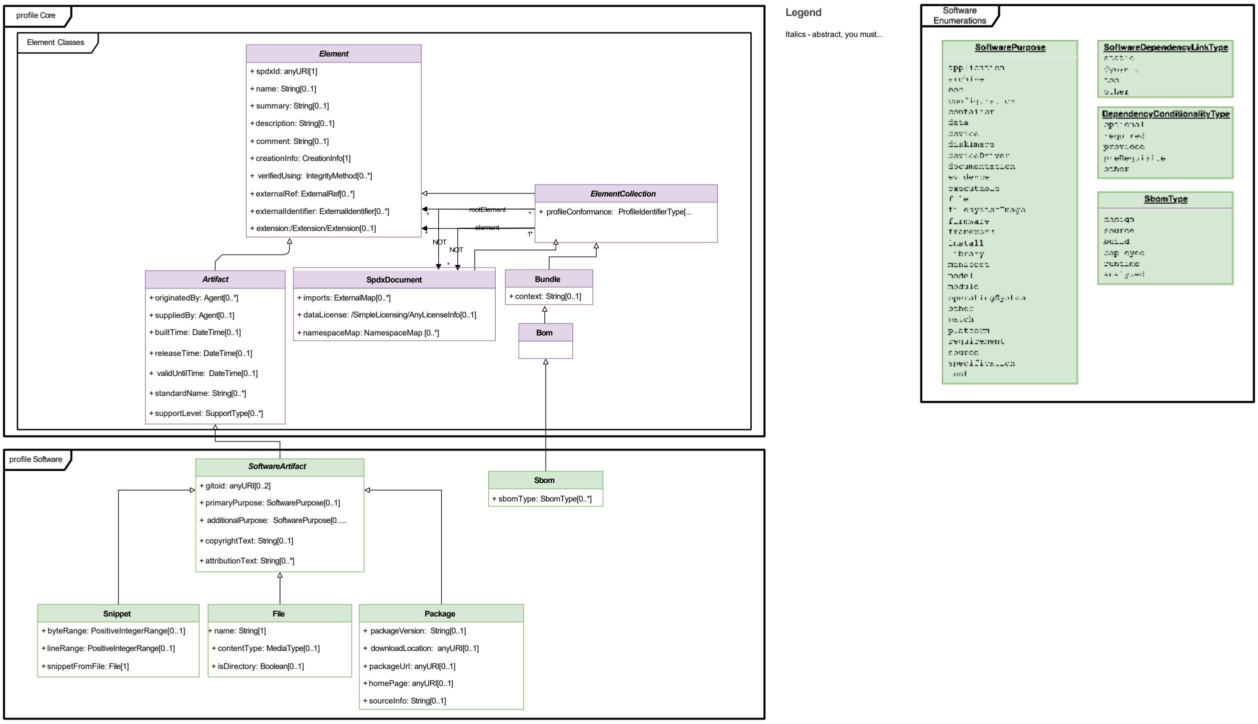

profile Core
Element Classes
Element
+ spdxId: anyURI[1]
+ name: String[0..1]
+ summary: String[0..1]
+ description: String[0..1]
+ comment: String[0..1]
+ creationInfo: CreationInfo[1]
+ verifiedUsing: IntegrityMethod[0..*]
+ externalRef: ExternalRef[0..*]
+ externalIdentifier: ExternalIdentifier[0..*]
+ extension:/Extension/Extension[0..1]
ElementCollection
+ profileConformance: ProfileIdentifierType[...
rootElement
element
*
1*
NOT
NOT
Artifact
+ originatedBy: Agent[0..*]
+ suppliedBy: Agent[0..1]
+ builtTime: DateTime[0..1]
+ releaseTime: DateTime[0..1]
+ validUntilTime: DateTime[0..1]
+ standardName: String[0..*]
+ supportLevel: SupportType[0..*]
SpdxDocument
+ imports: ExternalMap[0..*]
+ dataLicense: /SimpleLicensing/AnyLicenseInfo[0..1]
+ namespaceMap: NamespaceMap [0..*]
Bundle
+ context: String[0..1]
Bom
profile Software
SoftwareArtifact
+ gitoid: anyURI[0..2]
+ primaryPurpose: SoftwarePurpose[0..1]
+ additionalPurpose: SoftwarePurpose[0.....
+ copyrightText: String[0..1]
+ attributionText: String[0..*]
Sbom
+ sbomType: SbomType[0..*]
Snippet
+ byteRange: PositiveIntegerRange[0..1]
+ lineRange: PositiveIntegerRange[0..1]
+ snippetFromFile: File[1]
File
+ name: String[1]
+ contentType: MediaType[0..1]
+ isDirectory: Boolean[0..1]
Package
+ packageVersion: String[0..1]
+ downloadLocation: anyURI[0..1]
+ packageUrl: anyURI[0..1]
+ homePage: anyURI[0..1]
+ sourceInfo: String[0..1]
Legend
Italics - abstract, you must...
Software Enumerations
SoftwarePurpose
application
archive
bom
configuration
container
data
device
diskImage
deviceDriver
documentation
evidence
executable
file
filesystemImage
firmware
framework
install
library
manifest
model
module
operatingSystem
other
patch
platform
requirement
source
specification
test
SoftwareDependencyLinkType
static
dynamic
tool
other
DependencyConditionalityType
optional
required
provided
prerequisite
other
SbomType
design
source
build
deployed
runtime
analyzed

Figure 10 – Software Model profile, non-element classes, enumerations, & single data types

9.1 Software Profile Classes

9.1.1 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.

If the isDirectory property is specified and set to true, then the file represents a directory and all content stored in that directory.

Metadata

```
https://spdx.org/rdf/v3/Software/File
```

Name	File
Instantiability	Concrete
SubclassOf	/Software/SoftwareArtifact

Properties

Property	Type	minCount	maxCount
contentType	/Core/MediaType	0	1
isDirectory	xsd:boolean	0	1

9.1.2 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/v3/Software/Package
```

Name	Package
Instantiability	Concrete
SubclassOf	/Software/SoftwareArtifact

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

9.1.3 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/v3/Software/Sbom
```

Name	Sbom
Instantiability	Concrete
SubclassOf	/Core/Bom

Properties

Property	Type	minCount	maxCount
sbomType	SbomType	0	*

9.1.4 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/v3/Software/Snippet`

Name	Snippet
Instantiability	Concrete

SubclassOf	/Software/SoftwareArtifact

Properties

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

9.1.5 Software Artifact

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/v3/Software/SoftwareArtifact
```

Name	SoftwareArtifact
Instantiability	Abstract
SubclassOf	/Core/Artifact

Properties

Property	Type	minCount	maxCount
additionalPurpose	SoftwarePurpose	0	*
att ributionText	xsd:string	0	*
copyrightText	xsd:string	0	1
gitoid	xsd:anyURI	0	2
primaryPurpose	SoftwarePurpose	0	1

9.2 Software Profile Properties

9.2.1 additionalPurpose

Summary

Provides additional purpose information of the software artifact.

Description

Additional purpose provides information about the additional purposes of the software artifact in addition to the primaryPurpose.

Metadata

```
https://spdx.org/rdf/v3/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 att ributionText 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 licenseComment field) how they intend for data in this field to be used.

An attributionText is is not meant to include the software Package, File or Snippet's actual complete license text (see concludedLicense to identify the corresponding license).

Metadata

```
https://spdx.org/rdf/v3/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 W 3C pointer method vocabulary.

Metadata

```
https://spdx.org/rdf/v3/Software/byteRange
```

Name	byteRange
Nature	DataProperty
Range	/Core/PositiveIntegerRange

Referenced

- /Software/Snippet

9.2.4 contentType

Summary

Provides information about the content type of an Element.

Description

This field is a reasonable estimation of the content type of the Element, from a creator perspective. Content type is intrinsic to the Element, independent of how the Element is being used.

Metadata

```
https://spdx.org/rdf/v3/Software/contentType
```

Name	contentType
Nature	DataProperty
Range	/Core/MediaType

Referenced

- /Software/File

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

Metadata

```
https://spdx.org/rdf/v3/Software/copyrightText
```

Name	copyrightText

Nature	DataProperty
Range	xsd:string

Referenced

- /Software/SoftwareArtifact

9.2.6 downloadLocation

Summary

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

Description

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/v3/Software/downloadLocation
```

Name	downloadLocation
Nature	DataProperty
Range	xsd:anyURI

Referenced

- /Software/Package

9.2.7 gitoid

Summary

Used to record the artifact's gitoid: a canonical, unique, immutable identifier that can be used for software integrity verification.

Description

The gitoid is a canonical, unique, immutable artifact identifier for each software artifact. The gitoid for any software artifact can be calculated and recorded in SPDX 3.0 Snippet, File, or Package Elements.

The gitoid is defined as the Git Object Identifier of type expressed in blob of the software artifact, the gitoid URI scheme.

The OmniBOR ID for the OmniBOR Document associated with a software artifact should not be recorded in this field. Rather, OmniBOR IDs should be recorded in the SPDX Element's ExternalIdentifier property.

Metadata

```
https://spdx.org/rdf/v3/Software/gitoid
```

Name	gitoid
Nature	DataProperty
Range	xsd:anyURI

Referenced

- /Software/SoftwareArtifact

9.2.8 homePage

Summary

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

Description

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/v3/Software/homePage
```

Name	homePage
Nature	DataProperty
Range	xsd:anyURI

Referenced

- /Software/Package

9.2.9 isDirectory

Summary

If true, denotes the Element is a directory.

Description

If true, denotes the Element is a directory.

Metadata

```
https://spdx.org/rdf/v3/Software/isDirectory
```

Name	isDirectory
Nature	DataProperty
Range	xsd:boolean

Referenced

- /Software/File

9.2.10 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/v3/Software/lineRange
```

Name	lineRange
Nature	DataProperty
Range	/Core/PositiveIntegerRange

Referenced

- /Software/Snippet

9.2.11 packageUrl

Summary

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

Description

A packageUrl (commonly pronounced and referred to as "purl") is an att empt to standardize package representations in order to reliably identify and locate software packages. A purl 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.

the purl URL string is defined by seven components:

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

The definition for each component can be found in the purl specification. Components are designed such that they form a hierarchy from the most significant on the left to the least significant components on the right.

Parsing a purl string into its components works from left to right. Some extra type-specific normalizations are required. For more information, see How to parse a purl string in its components.

Metadata

```
https://spdx.org/rdf/v3/Software/packageUrl
```

Name	packageUrl
Nature	DataProperty
Range	xsd:anyURI

9.2.12 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/v3/Software/packageVersion
```

Name	packageVersion
Nature	DataProperty
Range	xsd:string

Referenced

- /Software/Package

9.2.13 primary Purpose

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/v3/Software/primaryPurpose
```

Name	primaryPurpose
Nature	ObjectProperty
Range	SoftwarePurpose

Referenced

- /Software/SoftwareArtifact

9.2.14 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.

Metadata

```
https://spdx.org/rdf/v3/Software/sbomType
```

Name	sbomType

Nature	ObjectProperty
Range	SbomType

Referenced

- /Software/Sbom

9.2.15 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/v3/Software/snippetFromFile
```

Name	snippetFromFile
Nature	ObjectProperty
Range	File

Referenced

- /Software/Snippet

9.2.16 sourceInfo

Summary

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

Description

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/v3/Software/sourceInfo
```

Name	sourceInfo
Nature	DataProperty
Range	xsd:string

Referenced

- /Software/Package

9.3 Software Profile Vocabularies

9.3.1 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, 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/v3/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.

9.3.2 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/v3/Software/SoftwarePurpose
```

Name	SoftwarePurpose

Entries

- application: the Element is a software application
- archive: the Element is an archived collection of one or more files (.tar, .zip, etc)
- bom: Element is a bill of materials
- configuration: Element is configuration data
- container: the Element is a container image which can be used by a container runtime application
- data: Element is data
- device: the Element refers to a chipset, processor, or electronic board
- deviceDriver: Element represents software that controls hardware devices
- diskImage: the Element refers to a disk image that can be writt en 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: Element is documentation
- evidence: the Element is the evidence that a specification or requirement has been fulfilled

10 Security Profile

Summary

The Security Profile captures security related information.

Description

The Security Profile captures security related information. Figure 7 below shows the logical model for the Security profile and its enumerations.

Metadata

Name	Security

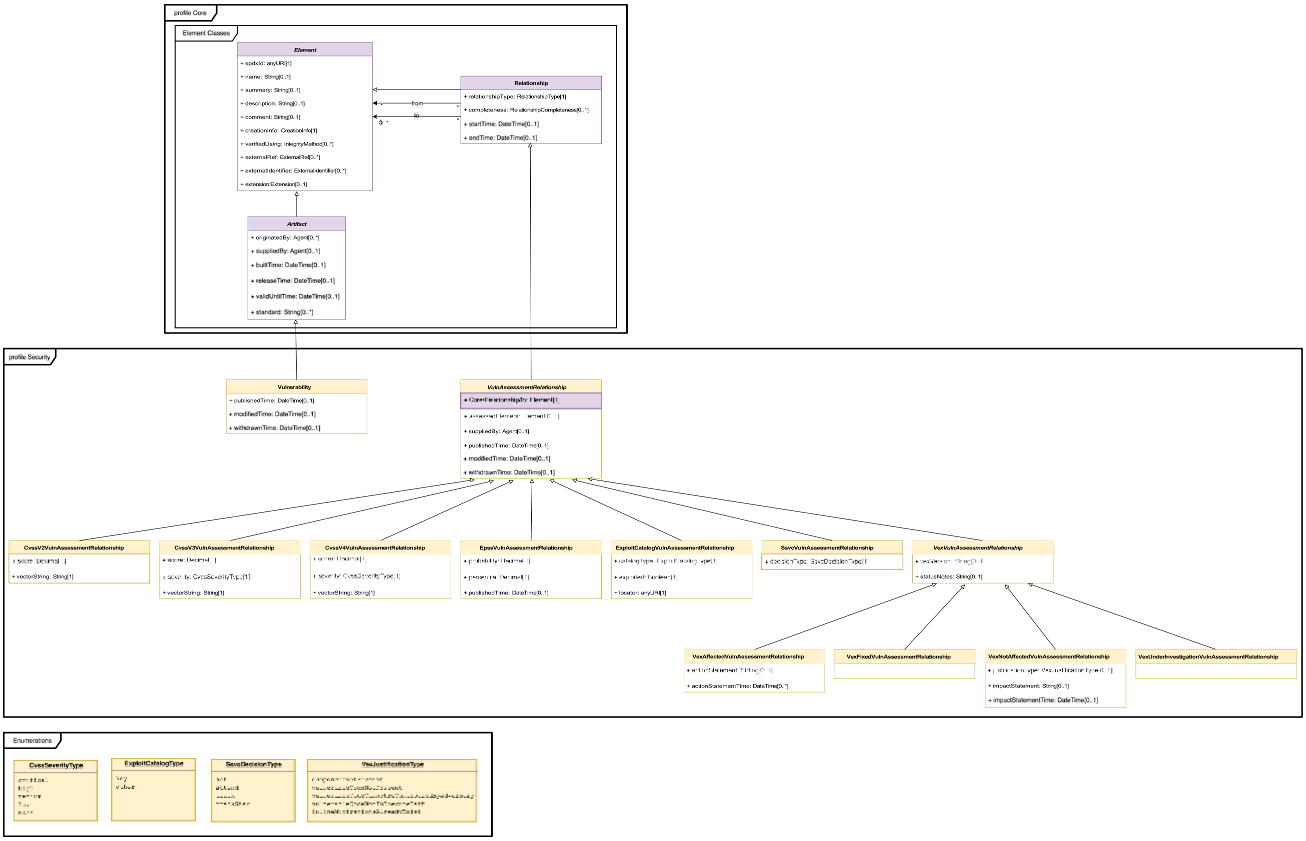

profile Core
Element Classes
Element
+ spdxId: anyURI[1]
+ name: String[0..1]
+ summary: String[0..1]
+ description: String[0..1]
+ comment: String[0..1]
+ creationInfo: CreationInfo[1]
+ verifiedUsing: IntegrityMethod[0..*]
+ externalRef: ExternalRef[0..*]
+ externalIdentifier: ExternalIdentifier[0..*]
+ extension:Extension[0..1]
Relationship
+ relationshipType: RelationshipType[1]
+ completeness: RelationshipCompleteness[0..1]
+ startTime: DateTime[0..1]
+ endTime: DateTime[0..1]
from
to
*
0..*
Artifact
+ originatedBy: Agent[0..*]
+ suppliedBy: Agent[0..1]
+ builtTime: DateTime[0..1]
+ releaseTime: DateTime[0..1]
+ validUntilTime: DateTime[0..1]
+ standard: String[0..*]
profile Security
Vulnerability
+ publishedTime: DateTime[0..1]
+ modifiedTime: DateTime[0..1]
+ withdrawnTime: DateTime[0..1]
VulnAssessmentRelationship
+ CoreRelationshipTo: Element[1]
+ assessedElement: Element[0..1]
+ suppliedBy: Agent[0..1]
+ publishedTime: DateTime[0..1]
+ modifiedTime: DateTime[0..1]
+ withdrawnTime: DateTime[0..1]
CvssV2VulnAssessmentRelationship
+ score: Decimal[1]
+ vectorString: String[1]
CvssV3VulnAssessmentRelationship
+ score: Decimal[1]
+ severity: CvssSeverityType[1]
+ vectorString: String[1]
CvssV4VulnAssessmentRelationship
+ score: Decimal[1]
+ severity: CvssSeverityType[1]
+ vectorString: String[1]
EpssVulnAssessmentRelationship
+ probability: Decimal[1]
+ percentile: Decimal[1]
+ publishedTime: DateTime[0..1]
ExploitCatalogVulnAssessmentRelationship
+ catalogType: ExploitCatalogType[1]
+ exploited: Boolean[1]
+ locator: anyURI[1]
SsvcVulnAssessmentRelationship
+ decisionType: SsvcDecisionType[1]
VexVulnAssessmentRelationship
+ vexVersion: String[0..1]
+ statusNotes: String[0..1]
VexAffectedVulnAssessmentRelationship
+ actionStatement: String[0..1]
+ actionStatementTime: DateTime[0..*]
VexFixedVulnAssessmentRelationship
VexNotAffectedVulnAssessmentRelationship
+ justificationType: VexJustificationType[0..1]
+ impactStatement: String[0..1]
+ impactStatementTime: DateTime[0..1]
VexUnderInvestigationVulnAssessmentRelationship
Enumerations
CvssSeverityType
critical
high
medium
low
none
ExploitCatalogType
kev
other
SsvcDecisionType
act
attend
track
trackStar
VexJustificationType
componentNotPresent
vulnerableCodeNotPresent
vulnerableCodeCannotBeControlledByAdversary
vulnerableCodeNotInExecutePath
inlineMitigationsAlreadyExist

Figure 11 – Security Model profile and enumerations

10.1 Security Profile 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 using version 2.0 of the Common Vulnerability Scoring System (CVSS) as defined at https://www.first.org/cvss/v2/guide. It is intended to communicate the results of using a CVSS calculator.

Constraints

- The relationship type must be set to hasAssessmentFor.

Syntax

```
{
  "@type": "CvssV2VulnAssessmentRelationship",
  "@id": "urn:spdx.dev:cvssv2-cve-2020-28498",
  "relationshipType": "hasAssessmentFor",
  "score": 4.3,
  "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"],
  "assessedElement": "urn:npm-elliptic-6.5.2",
  "externalRefs": [
    {
      "@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",
  "@id": "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/v3/Security/CvssV2VulnAssessmentRelationship

Name	CvssV2VulnAssessmentRelationship
Instantiability	Concrete
SubclassOf	VulnAssessmentRelationship

Properties

Property	Type	minCount	maxCount
score	xsd:decimal	1	1
vectorString	xsd:string	1	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 using version 3.0 or 3.1 of the Common Vulnerability Scoring System (CVSS). It is intended to communicate the results of using a CVSS calculator.

Constraints

- The value of severity must be one of 'NONE', 'LOW ', 'MEDIUM', 'HIGH' or 'CRITICAL'.
- The relationship type must be set to hasAssessmentFor.

Syntax

```
{
 "@type":"CvssV3VulnAssessmentRelationship",
 "@id":"urn:spdx.dev:cvssv3-cve-2020-28498",
 "relationshipType":"hasAssessmentFor",
 "score":6.8,
 "severity":"MEDIUM",
 "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"],
 "assessedElement":"urn:npm-elliptic-6.5.2",
 "externalRefs":[
  {
   "@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",
 "@id":"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/v3/Security/CvssV3VulnAssessmentRelationship`

Name	CvssV3VulnAssessmentRelationship
Instantiability	Concrete
SubclassOf	VulnAssessmentRelationship

Properties

Property	Type	minCount	maxCount
score	xsd:decimal	1	1
severity	CvssSeverityType	1	1
vectorString	xsd:string	1	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 using version 4 of the Common Vulnerability Scoring System (CVSS) as defined on https://www.first.org/cvss/v4.0/specification-document. It is intented to communicate the results of using a CVSS calculator.

Constraints

- The value of severity must be one of 'NONE', 'LOW ', 'MEDIUM', 'HIGH' or 'CRITICAL'.
- The relationship type must be set to hasAssessmentFor.

Syntax

```
{
 "@type": "CvssV4VulnAssessmentRelationship",
 "@id": "urn:spdx.dev:cvssv4-cve-2021-44228",
 "relationshipType": "hasAssessmentFor",
 "severity": "MEDIUM",
 "score": 10.0,
 "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"],
 "assessedElement": "urn:apache-log4j-2.14.1",
 "externalRefs": [
   {
    "@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-JNDI-Command-Execution-log4shell-Vulnerability-CVE-2021-44228"
   },
 ],
 "suppliedBy": ["urn:spdx.dev:agent-my-security-vendor"],
 "publishedTime": "2023-10-05T23:09:13Z"
},

{
 "@type": "Relationship",
 "@id": "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/v3/Security/CvssV4VulnAssessmentRelationship`

Name	CvssV4VulnAssessmentRelationship
Instantiability	Concrete
SubclassOf	VulnAssessmentRelationship

Properties

Property	Type	minCount	maxCount
score	xsd:decimal	1	1
severity	CvssSeverityType	1	1
vectorString	xsd:string	1	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 using the Exploit Prediction Scoring System (EPSS) as defined at https://www.first.org/epss/model.

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.

Syntax

```
{
 "@type": "EpssVulnAssessmentRelationship",
 "@id": "urn:spdx.dev:epss-CVE-2020-28498",
 "relationshipType": "hasAssessmentFor",
 "probability": 0.00105,
 "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"
}
```

Metadata

`https://spdx.org/rdf/v3/Security/EpssVulnAssessmentRelationship`

Name	EpssVulnAssessmentRelationship
Instantiability	Concrete

SubclassOf	VulnAssessmentRelationship

Properties

Property	Type	minCount	maxCount
percentile	xsd:decimal	1	1
probability	xsd:decimal	1	1
publishedTime	/Core/DateTime	1	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 Catalog (KEV) https://www.cisa.gov/known-exploited-vulnerabilities-catalog.

Constraints

- The relationship type must be set to hasAssessmentFor.

Syntax

```
{
 "@type": "ExploitCatalogVulnAssessmentRelationship",
 "@id": "urn:spdx.dev:exploit-catalog-1",
 "relationshipType": "hasAssessmentFor",
 "catalogType": "kev",
 "locator": "https://www.cisa.gov/known-exploited-vulnerabilities-catalog",
 "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/v3/Security/ExploitCatalogVulnAssessmentRelationship`

Name	ExploitCatalogVulnAssessmentRelationship
Instantiability	Concrete
SubclassOf	VulnAssessmentRelationship

Properties

Property	Type	minCount	maxCount
catalogType	ExploitCatalogType	1	1
exploited	xsd:boolean	1	1
locator	xsd:anyURI	1	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 on https://www.cisa.gov/stakeholder-specific-vulnerability-categorization-ssvc. It is intended to communicate the results of using the CISA SSVC Calculator.

Constraints

- The relationship type must be set to hasAssessmentFor.

Syntax

```
{
 "@type": "SsvcVulnAssessmentRelationship",
 "@id": "urn:spdx.dev:ssvc-1",
 "relationshipType": "hasAssessmentFor",
 "decisionType": "act",
 "from": "urn:spdx.dev:vuln-cve-2020-28498",
 "to": ["urn:product-acme-application-1.3"],
 "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/v3/Security/SsvcVulnAssessmentRelationship
```

Name	SsvcVulnAssessmentRelationship
Instantiability	Concrete
SubclassOf	VulnAssessmentRelationship

Properties

Property	Type	minCount	maxCount
decisionType	SsvcDecisionType	1	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.

Syntax

```
{
 "@type": "VexAffectedVulnAssessmentRelationship",
 "@id": "urn:spdx.dev:vex-affected-1",
 "relationshipType": "affects",
 "from":  "urn:spdx.dev:vuln-cve-2020-28498",

 "to": ["urn:product-acme-application-1.3"],
 "assessedElement": "urn:npm-elliptic-6.5.2",
 "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/v3/Security/VexAffectedVulnAssessmentRelationship`

Name	VexAffectedVulnAssessmentRelationship
Instantiability	Concrete
SubclassOf	VexVulnAssessmentRelationship

Properties

Property	Type	minCount	maxCount
actionStatement	xsd:string	0	1
actionStatementTime	/Core/DateTime	0	*

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 ve a /Security/Vulnerability classed element.

Syntax

```
{
 "@type": "VexFixedVulnAssessmentRelationship",
 "@id": "urn:spdx.dev:vex-fixed-in-1",
 "relationshipType": "fixedIn",
 "from": "urn:spdx.dev:vuln-cve-2020-28498",
 "to": ["urn:product-acme-application-1.3"],
 "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/v3/Security/VexFixedVulnAssessmentRelationship`

Name	VexAffectedVulnAssessmentRelationship

Instantiability	Concrete
SubclassOf	VexVulnAssessmentRelationship

Properties

Property	Type	minCount	maxCount
actionStatement	xsd:string	0	1
actionStatementTime	/Core/DateTime	0	*

10.1.9 Vex NotAffectedVulnAssessmentRelationship

Summary

Links a vulnerability and one or more elements designating the latt er 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.

Syntax

```
{
  "@type": "VexNotAffectedVulnAssessmentRelationship",
  "@id": "urn:spdx.dev:vex-not-affected-1",
  "relationshipType": "doesNotAffect",
  "from": "urn:spdx.dev:vuln-cve-2020-28498",
  "to": ["urn:product-acme-application-1.3"],
  "assessedElement": "urn:npm-elliptic-6.5.2",
  "justificationType": "componentNotPresent",
  "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/v3/Security/VexFixedVulnAssessmentRelationship

Name	VexFixedVulnAssessmentRelationship
Instantiability	Concrete
SubclassOf	VexVulnAssessmentRelationship

Properties

Property	Type	minCount	maxCount
impactStatement	xsd:string	0	1
impactStatementTime	/Core/DateTime	0	1
justificationType	VexJustificationType	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 ve a /Security/Vulnerability classed element.

Syntax

```
{
 "@type": "VexUnderInvestigationVulnAssessmentRelationship",
 "@id": "urn:spdx.dev:vex-underInvestigation-1",
 "relationshipType": "underInvestigationFor",
 "from": "urn:spdx.dev:vuln-cve-2020-28498",
 "to": ["urn:product-acme-application-1.3"],
 "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/v3/Security/VexUnderInvestigationVulnAssessmentRelationship`

Name	VexNotAffectedVulnAssessmentRelationship
Instantiability	Concrete
SubclassOf	VexVulnAssessmentRelationship

Properties

Property	Type	minCount	maxCount

10.1.11 VexVulnAssessmentRelationship

Summary

Asbtract 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/v3/Security/VexVulnAssessmentRelationship
```

Name	VexVulnAssessmentRelationship
Instantiability	Abstract
SubclassOf	VulnAssessmentRelationship

Properties

Property	Type	minCount	maxCount
statusNotes	xsd:string	0	1
vexVersion	xsd:string	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/v3/Security/VulnAssessmentRelationship
```

Name	VulnAssessmentRelationship
Instantiability	Abstract
SubclassOf	/Core/Relationship

Properties

Property	Type	minCount	maxCount
/Core/suppliedBy	/Core/Agent	0	1
assessedElement	/Core/Element	0	1
modifiedTime	/Core/DateTime	0	1
publishedTime	/Core/DateTime	0	1
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.

Syntax

```
{
  "@type": "Vulnerability",
  "@id": "urn:spdx.dev:vuln-1",
  "summary": "Use of a Broken or Risky Cryptographic Algorithm",
  "description": "The npm package `elliptic` before version 6.5.4 are vulnerable to Cryptographic Issues via the secp256k1 implementation in
elliptic/ec/key.js. There is no check to confirm that the public key point passed into the derive function actually exists on the secp256k1 curve. This
results in the potential for the private key used in this implementation to be revealed after a number of ECDH operations are performed.",
"modified": "2021-03-08T16:02:43Z",
  "published":  "2021-03-08T16:06:50Z",
  "externalIdentifiers":  [
    {
      "@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"
    }
  ],
  "externalRefs": [
    {
        "@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/blob/master/2020_05_26_secp256k1_twist_attacks/secp256k1_twist_attacks.md"
    }
  ]
},
{
  "@type": "Relationship",
  "@id": "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",

  "@id": "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/v3/Security/Vulnerability`

Name	Vulnerability
Instantiability	Concrete
SubclassOf	/Core/Artifact

Properties

Property	Type	minCount	maxCount
modifiedTime	/Core/DateTime	0	1
publishedTime	/Core/DateTime	0	1
withdrawnTime	/Core/DateTime	0	1

10.2 Security Profile 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/v3/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/v3/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/v3/Security/assessedElement
```

Name	assessedElement
Nature	ObjectProperty
Range	/Core/Element

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.md vocabulary.

Metadata

```
https://spdx.org/rdf/v3/Security/catalogType
```

Name	catalogType
Nature	ObjectProperty
Range	ExploitCatalogType

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/sites/default/files/publications/cisa-ssvc-guide%20508c.pdf

Description

A decisionType is a mandatory value and must select one of the four entries in the vocabulary.

Metadata

```
https://spdx.org/rdf/v3/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/v3/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 prouct(s) are not affected by the vulnerability must be provided.

Metadata

```
https://spdx.org/rdf/v3/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/v3/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/v3/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/v3/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/v3/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 EPSS score. https://www.first.org/epss/data_stats

Metadata

```
https://spdx.org/rdf/v3/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.

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). https://www.first.org/epss/data_stats

Metadata

```
https://spdx.org/rdf/v3/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/v3/Security/publishedTime
```

Name	publishedTime
Nature	DataProperty
Range	/Core/DateTime

Referenced

- /Security/EpssVulnAssessmentRelationship
- /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 on https://www.first.org/cvss.

Metadata

```
https://spdx.org/rdf/v3/Security/score
```

Name	score
Nature	DataProperty
Range	xsd:decimal

Referenced

- /Security/CvssV2VulnAssessmentRelationship
- /Security/CvssV3VulnAssessmentRelationship
- /Security/CvssV4VulnAssessmentRelationship

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/v3/Security/severity
```

Name	severity

Nature	DataProperty
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/v3/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. https://www.first.org/cvss/v4.0/specification- document#Vector-String

Metadata

```
https://spdx.org/rdf/v3/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 the VEX document.

Description

The document version default value is zero. When any VEX-related content changes, the version must be incremented.

Metadata

```
https://spdx.org/rdf/v3/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/v3/Security/withdrawnTime
```

Name	withdrawnTime
Nature	DataProperty
Range	/Core/DateTime

Referenced

- /Security/VulnAssessmentRelationship
- /Security/Vulnerability

10.3 Security Profile Vocabularies

10.3.1 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 using version 2.0 of the Common Vulnerability Scoring System (CVSS) as defined at https://www.first.org/cvss/v2/guide. It is intended to communicate the results of using a CVSS calculator.

Constraints

- The relationship type must be set to hasAssessmentFor.

Syntax

10.3.2 CvssSeverityType

Summary

Specifies the CVSS base, temporal, threat, or environmental severity type.

Description

CvssSeverityType specifies the 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 CVSS versions 3 and 4. CvssSeverityType is a mandatory field because baseSeverity is required in the CVSS version 3.0, 3.1, and 4.0 schemas. The field can be used to document the base, temporal, threat, or environmental severity.

Metadata

```
https://spdx.org/rdf/v3/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 - 3.9
- medium: When a CVSS score is between 4 - 6.9
- none: When a CVSS score is 0

10.3.3 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/v3/Security/ExploitCatalogType`

Name	ExploitCatalogType

Entries

- kev: CISA's Known Exploited Vulnerability (KEV) Catalog
- other: Other exploit catalogs

10.3.4 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) system https://www.cisa.gov/stakeholder-specific-vulnerability-categorization-ssvc

Metadata

`https://spdx.org/rdf/v3/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.

10.3.5 VexJustificationType

Summary

Specifies the VEX justification type.

Description

VexJustificationType specifies the type of Vulnerability Exploitability eXchange (VEX) justification.

Metadata

`https://spdx.org/rdf/v3/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 att acker cannot mount any anticipated att ack.
- 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 Profile

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 concludedLicense Relationship.

Classes and Property restrictions are defined in the SimpleLicensingProfile (Classes and Properties associated with string license expressions) and in the ExpandedLicensingProfile (Classes and Properties used for a fully parsed syntax tree of license expressions).

There are 2 relationship types related to licensing - declaredLicense and concludedLicense.

A declaredLicense 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 concludedLicense field.

A declaredLicense may be expressed differently in practice for different types of Software Artifacts. For example:

- for Packages:
- 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.)
- 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:
- would include license info found in the File itself (e.g., license header or notice, comments indicating the license, SPDX-License-Identifier expression)
- would not include license info found in a different file (e.g., LICENSE file in the top directory of a repository)
- for Snippets:
- would include license info found in the Snippet itself (e.g., license notice, comments, SPDX-License-Identifier expression)
- 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 declaredLicense relationship to NoneLicense indicates that the corresponding Package, File or Snippet contains no license information whatsoever.

A declaredLicense 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 declaredLicense relationship is not present, no assumptions can be made about whether or not a declaredLicense exists. Note that a missing declaredLicense 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 declaredLicense relationship.

A concludedLicense 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 concludedLicense relationship to NoneLicense indicates that the SPDX data creator has looked and did not find any license information for this Software Artifact.

A concludedLicense 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 concludedLicense is not present, no assumptions can be made about whether or not a concludedLicense exists. Note that a missing concludedLicense 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 concludedLicense relationship.

A written explanation of a relationship to a NoAssertionLicense MAY be provided in the comment field for the relationship.

If the concludedLicense for a Software Artifact is not the same as its declaredLicense, a written explanation SHOULD be provided in the concludedLicense relationship comment field.

Figure 8 below shows the logical model for the Simple and Expanded Licensing profiles.

Metadata

Name	Licensing

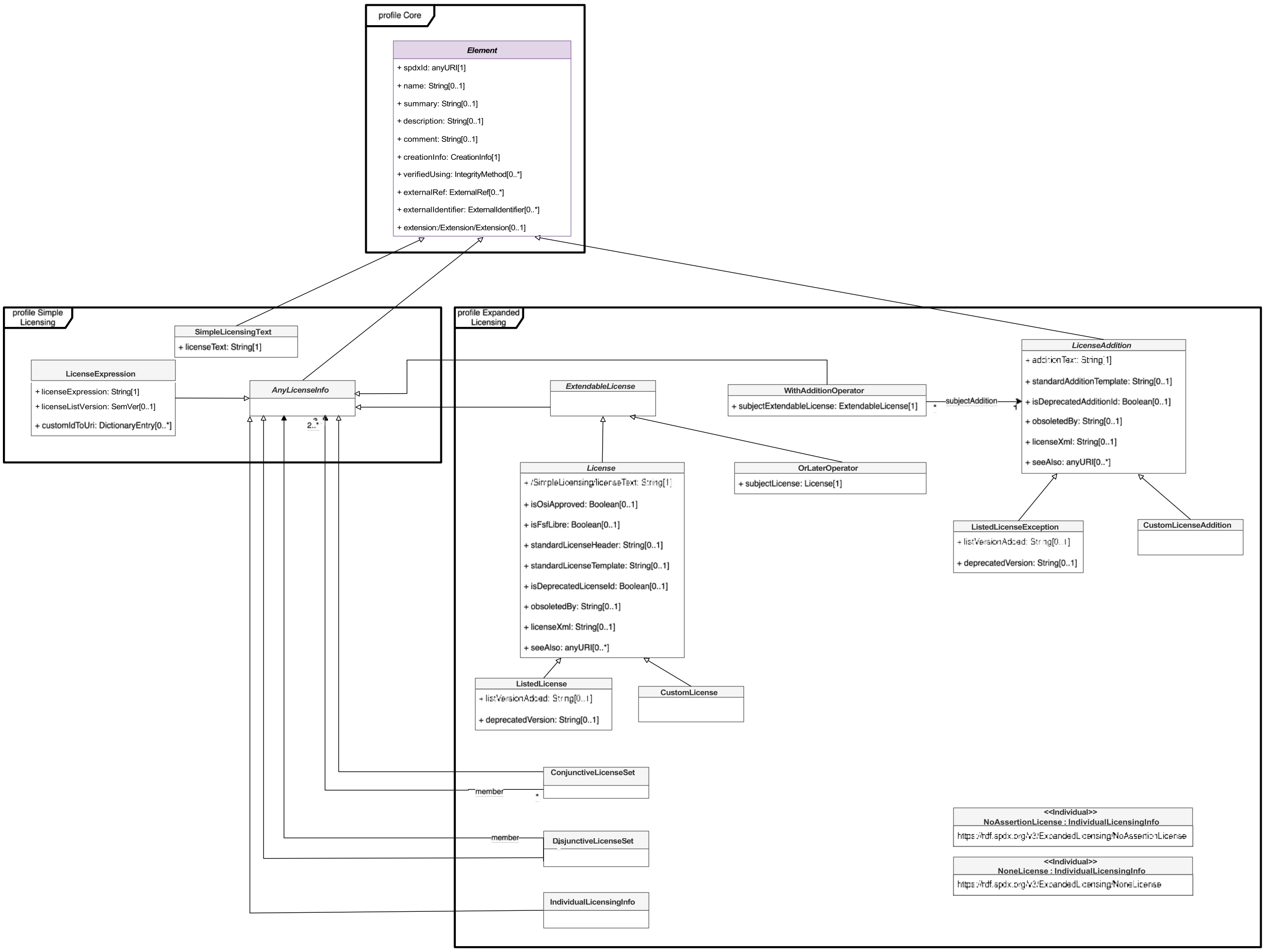

profile Core
Element
+ spdxId: anyURI[1]
+ name: String[0..1]
+ summary: String[0..1]
+ description: String[0..1]
+ comment: String[0..1]
+ creationInfo: CreationInfo[1]
+ verifiedUsing: IntegrityMethod[0..*]
+ externalRef: ExternalRef[0..*]
+ externalIdentifier: ExternalIdentifier[0..*]
+ extension:/Extension/Extension[0..1]
profile Simple Licensing
SimpleLicensingText
+ licenseText: String[1]
LicenseExpression
+ licenseExpression: String[1]
+ licenseListVersion: SemVer[0..1]
+ customIdToUri: DictionaryEntry[0..*]
AnyLicenseInfo
2..*
profile Expanded Licensing
ExtendableLicense
WithAdditionOperator
+ subjectExtendableLicense: ExtendableLicense[1]
*
subjectAddition
1
LicenseAddition
+ additionText: String[1]
+ standardAdditionTemplate: String[0..1]
+ isDeprecatedAdditionId: Boolean[0..1]
+ obsoletedBy: String[0..1]
+ licenseXml: String[0..1]
+ seeAlso: anyURI[0..*]
License
+ /SimpleLicensing/licenseText: String[1]
+ isOsiApproved: Boolean[0..1]
+ isFsfLibre: Boolean[0..1]
+ standardLicenseHeader: String[0..1]
+ standardLicenseTemplate: String[0..1]
+ isDeprecatedLicenseId: Boolean[0..1]
+ obsoletedBy: String[0..1]
+ licenseXml: String[0..1]
+ seeAlso: anyURI[0..*]
OrLaterOperator
+ subjectLicense: License[1]
ListedLicenseException
+ listVersionAdded: String[0..1]
+ deprecatedVersion: String[0..1]
CustomLicenseAddition
ListedLicense
+ listVersionAdded: String[0..1]
+ deprecatedVersion: String[0..1]
CustomLicense
ConjunctiveLicenseSet
member
*
DisjunctiveLicenseSet
member
IndividualLicensingInfo
<<Individual>>
NoAssertionLicense : IndividualLicensingInfo
https://rdf.spdx.org/v3/ExpandedLicensing/NoAssertionLicense
<<Individual>>
NoneLicense : IndividualLicensingInfo
https://rdf.spdx.org/v3/ExpandedLicensing/NoneLicense

Figure 12 – Licensing Simple and Expanded Model profiles

11.1 SimpleLicensing Profile

Summary

Additional metadata relating to software licensing.

Description

The SimpleLicensing profile provides classes and properties to express licenses as a license expression 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.

The ExpandingLicensing profile can be used to represent the complete parsed license expressions.

Metadata

```
https://spdx.org/rdf/v3/SimpleLicensing
```

Name	SimpleLicensing

SimpleLicensing Classes

11.1.1 AnyLicenseInfo

Summary

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.

Description

An AnyLicenseInfo is used by licensing properties of software artifacts. It can be a NoneLicense, a NoAssertionLicense, single license (either on the SPDX License List or a custom-defined license); 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/v3/SimpleLicensing/AnyLicenseInfo
```

Name	AnyLicenseInfo
Instantiability	Abstract
SubclassOf	/Core/Element

Properties

Property	Type	minCount	maxCount

11.1.2 LicenseExpression

Summary

An SPDX Element containing an SPDX license expression string.

Description

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 an SPDX License Expression in this section.

Metadata

```
https://spdx.org/rdf/v3/SimpleLicensing/LicenseExpression
```

Name	LicenseExpression
Instantiability	Concrete
SubclassOf	AnyLicenseInfo

Properties

Property	Type	minCount	maxCount
customIdToUri	/Core/DictionaryEntry	0	*
licenseExpression	xsd:string	1	1
licenseListVersion	/Core/SemVer	0	1

11.1.3 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 at https://spdx.org/licenses, and is therefore defined by an SPDX data creator.

Metadata

```
https://spdx.org/rdf/v3/SimpleLicensing/SimpleLicensingText
```

Name	SimpleLicensingText

Instantiability	Concrete
SubclassOf	/Core/Element

Properties

Property	Type	minCount	maxCount
licenseText	xsd:string	1	1

SimpleLicensing Properties

11.1.4 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 dictates any refence starting with a "LicenseRef-" or "AdditionRef-" refers to license or addition text not found in the SPDX list of licenses. These custom licenses must be a CustomLicense, a CustomLicenseAddtion, 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 corrosponding CustomLicense, CustomLicenseAddition, or SimpleLicensingText.

Metadata

```
https://spdx.org/rdf/v3/SimpleLicensing/customIdToUri
```

Name	customIdToUri
Nature	ObjectProperty
Range	/Core/DictionaryEntry

Referenced

- /SimpleLicensing/LicenseExpression

11.1.5 licenseExpression

Summary

A string in the license expression format.

Description

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 an SPDX License Expression in this section.

Metadata

```
https://spdx.org/rdf/v3/SimpleLicensing/licenseExpression
```

Name	licenseExpression
Nature	DataProperty
Range	xsd:string

Referenced

- /SimpleLicensing/LicenseExpression

11.1.6 licenseListVersion

Summary

The version of the SPDX License List used in the license expression.

Description

Recognizing that licenses are added to the SPDX License List 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/v3/SimpleLicensing/licenseListVersion
```

Name	licenseListVersion
Nature	DataProperty
Range	/Core/SemVer

Referenced

- /SimpleLicensing/LicenseExpression

11.1.7 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 Matching Guidelines when comparing it to another text for matching purposes.

Metadata

```
https://spdx.org/rdf/v3/SimpleLicensing/licenseText
```

Name	licenseText
Nature	DataProperty
Range	xsd:string

Referenced

- /ExpandedLicensing/License
- /SimpleLicensing/SimpleLicensingText

11.2 ExpandedLicensing Profile

Summary

Fully expanded license expressions.

Description

This profile supports representing a fully expanded license expression in object form.

Metadata

```
https://spdx.org/rdf/v3/ExpandedLicensing
```

Name	ExpandedLicensing

ExpandedLicensing Classes

11.2.1 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 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/v3/ExpandedLicensing/ConjunctiveLicenseSet
```

Name	ConjunctiveLicenseSet
Instantiability	Concrete
SubclassOf	/SimpleLicensing/AnyLicenseInfo

Properties

Property	Type	minCount	maxCount
member	/SimpleLicensing/AnyLicenseInfo	2	*

11.2.2 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 at https://spdx.org/licenses, and is therefore defined by an SPDX data creator.

Metadata

```
https://spdx.org/rdf/v3/ExpandedLicensing/CustomLicense
```

Name	CustomLicense
Instantiability	Concrete
SubclassOf	License

Properties

Property	Type	minCount	maxCount

11.2.3 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 Exceptions List at https://spdx.org/licenses/exceptions-index.html, 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.

Metadata

```
https://spdx.org/rdf/v3/ExpandedLicensing/CustomLicenseAddition
```

Name	CustomLicenseAddition
Instantiability	Concrete
SubclassOf	LicenseAddition

Properties

Property	Type	minCount	maxCount

11.2.4 DisjunctiveLicenseSet

Summary

Portion of an AnyLicenseInfo representing a set of licensing information where only any 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/v3/ExpandedLicensing/DisjunctiveLicenseSet
```

Name	DisjunctiveLicenseSet
Instantiability	Concrete
SubclassOf	/SimpleLicensing/AnyLicenseInfo

Properties

Property	Type	minCount	maxCount
Member	/SimpleLicensing/AnyLicenseInfo	2	*

11.2.5 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/v3/ExpandedLicensing/ExtendableLicense
```

Name	ExtendableLicense
Instantiability	Abstract
SubclassOf	/SimpleLicensing/AnyLicenseInfo

Properties

Property	Type	minCount	maxCount

11.2.6 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/v3/ExpandedLicensing/IndividualLicensingInfo
```

Name	IndividualLicensingInfo
Instantiability	Concrete
SubclassOf	/SimpleLicensing/AnyLicenseInfo

Properties

Property	Type	minCount	maxCount

11.2.7 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 (ListedLicense) or defined by an SPDX data creator (CustomLicense).

Metadata

```
https://spdx.org/rdf/v3/ExpandedLicensing/License
```

Name	License
Instantiability	Abstract
SubclassOf	ExtendableLicense

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

11.2.8 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 Exceptions List (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/v3/ExpandedLicensing/LicenseAddition`

Name	LicenseAddition
Instantiability	Abstract
SubclassOf	/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

11.2.9 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 at https://spdx.org/licenses.

Metadata

```
https://spdx.org/rdf/v3/ExpandedLicensing/ListedLicense
```

Name	ListedLicense
Instantiability	Concrete
SubclassOf	License

Properties

Property	Type	minCount	maxCount
deprecatedVersion	xsd:string	0	1
listVersionAdded	xsd:string	0	1

11.2.10 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 Exceptions List at https://spdx.org/licenses/exceptions-index.html.

Metadata

```
https://spdx.org/rdf/v3/ExpandedLicensing/ListedLicenseException
```

Name	ListedLicenseException
Instantiability	Concrete
SubclassOf	LicenseAddition

Properties

Property	Type	minCount	maxCount
deprecatedVersion	xsd:string	0	1
listVersionAdded	xsd:string	0	1

11.2.11 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/v3/ExpandedLicensing/OrLaterOperator
```

Name	OrLaterOperator
Instantiability	Concrete
SubclassOf	ExtendableLicense

Properties

Property	Type	minCount	maxCount
subjectLicense	License	1	1

11.2.12 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 Exceptions List (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/v3/ExpandedLicensing/WithAdditionOperator
```

Name	WithAdditionOperator
Instantiability	Concrete
SubclassOf	/SimpleLicensing/AnyLicenseInfo

Properties

Property	Type	minCount	maxCount
subjectAddition	LicenseAddition	1	1
subjectExtendableLicense	ExtendableLicense	1	1

ExpandedLicensing Properties

11.2.13 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 Matching Guidelines when comparing it to another text for matching purposes.

Metadata

```
https://spdx.org/rdf/v3/ExpandedLicensing/additionText
```

Name	additionText
Nature	DataProperty
Range	xsd:string

Referenced

- /ExpandedLicensing/LicenseAddition

11.2.14 deprecatedVersion

Summary

Specifies the SPDX License List version in which this license or exception identifier was deprecated.

Description

A deprecatedVersion for a ListedLicense or ListedLicenseException on the SPDX License List specifies which version release of the License List was the first one in which it was marked as deprecated.

Metadata

```
https://spdx.org/rdf/v3/ExpandedLicensing/deprecatedVersion
```

Name	deprecatedVersion
Nature	DataProperty
Range	xsd:string

Referenced

- /ExpandedLicensing/ListedLicense
- /ExpandedLicensing/ListedLicenseException

11.2.15 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 Exceptions List, then the deprecatedVersion property indicates on which version release of the Exceptions List it was first marked as deprecated.

"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/v3/ExpandedLicensing/isDeprecatedAdditionId`

Name	isDeprecatedAdditionId
Nature	DataProperty
Range	xsd:boolean

Referenced

- /ExpandedLicensing/LicenseAddition

11.2.16 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, 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/v3/ExpandedLicensing/isDeprecatedLicenseId
```

Name	isDeprecatedLicenseId
Nature	DataProperty
Range	xsd:boolean

Referenced

- /ExpandedLicensing/License

11.2.17 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 has listed this License as "free" in their commentary on licenses, located at the time of this writing at https://www.gnu.org/licenses/license-list.en.html.

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.

Metadata

```
https://spdx.org/rdf/v3/ExpandedLicensing/isFsfLibre
```

Name	isFsfLibre
Nature	DataProperty
Range	xsd:boolean

Referenced

- /ExpandedLicensing/License

11.2.18 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) has listed this License as "approved" in their list of OSI Approved Licenses, located at the time of this writing at https://opensource.org/licenses/.

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/v3/ExpandedLicensing/isOsiApproved
```

Name	isOsiApproved
Nature	DataProperty
Range	xsd:boolean

Referenced

- /ExpandedLicensing/License

11.2.19 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. See the XML fields defined at https://github.com/spdx/license-list-XML/blob/main/DOCS/xml-fields.md for a text description. There is also an XML schema available at https://github.com/spdx/license-list-XML/blob/main/schema/ListedLicense.xsd.

Metadata

```
https://spdx.org/rdf/v3/ExpandedLicensing/licenseXml
```

Name	licenseXml
Nature	DataProperty
Range	xsd:string

Referenced

- /ExpandedLicensing/License
- /ExpandedLicensing/LicenseAddition

11.2.20 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 specifies which version release of the License List was the first one in which it was included.

Metadata

```
https://spdx.org/rdf/v3/ExpandedLicensing/listVersionAdded
```

Name	listVersionAdded
Nature	DataProperty
Range	xsd:string

Referenced

- /ExpandedLicensing/ListedLicense
- /ExpandedLicensing/ListedLicenseException

11.2.21 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/v3/ExpandedLicensing/member
```

Name	member
Nature	ObjectProperty
Range	/SimpleLicensing/AnyLicenseInfo

Referenced

- /ExpandedLicensing/ConjunctiveLicenseSet
- /ExpandedLicensing/DisjunctiveLicenseSet

11.2.22 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.

Metadata

`https://spdx.org/rdf/v3/ExpandedLicensing/obsoletedBy`

Name	obsoletedBy
Nature	DataProperty
Range	xsd:string

Referenced

- /ExpandedLicensing/License
- /ExpandedLicensing/LicenseAddition

11.2.23 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/v3/ExpandedLicensing/seeAlso`

Name	seeAlso
Nature	DataProperty
Range	xsd:anyURI

Referenced

- /ExpandedLicensing/License
- /ExpandedLicensing/LicenseAddition

11.2.24 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 specification for format information.

Metadata

```
https://spdx.org/rdf/v3/ExpandedLicensing/standardAdditionTemplate
```

Name	standardAdditionTemplate
Nature	DataProperty
Range	xsd:string

Referenced

- /ExpandedLicensing/LicenseAddition

11.2.25 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/v3/ExpandedLicensing/standardLicenseHeader
```

Name	standardLicenseHeader
Nature	DataProperty
Range	xsd:string

Referenced

- /ExpandedLicensing/License

11.2.26 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 specification for format information.

Metadata

`https://spdx.org/rdf/v3/ExpandedLicensing/standardLicenseTemplate`

Name	standardLicenseTemplate
Nature	DataProperty
Range	xsd:string

Referenced

- /ExpandedLicensing/License

11.2.27 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).

Metadata

`https://spdx.org/rdf/v3/ExpandedLicensing/subjectAddition`

Name	subjectAddition
Nature	ObjectProperty
Range	LicenseAddition

Referenced

- /ExpandedLicensing/WithAdditionOperator

11.2.28 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/v3/ExpandedLicensing/subjectExtendableLicense
```

Name	subjectExtendableLicense
Nature	ObjectProperty
Range	ExtendableLicense

Referenced

- /ExpandedLicensing/WithAdditionOperator

11.2.29 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/v3/ExpandedLicensing/subjectLicense
```

Name	subjectLicense
Nature	ObjectProperty
Range	License

Referenced

- /ExpandedLicensing/OrLaterOperator

ExpandedLicensing Individuals

11.2.30 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/v3/ExpandedLicensing/NoAssertionLicense
```

Name	NoAssertionLicense
Type	IndividualLicensingInfo
IRI	https://spdx.org/rdf/v3/Licensing/NoAssertion

11.2.31 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/v3/ExpandedLicensing/NoneLicense
```

Name	NoneLicense
Type	IndividualLicensingInfo
IRI	https://spdx.org/rdf/v3/Licensing/None

12 Dataset Profile

Summary

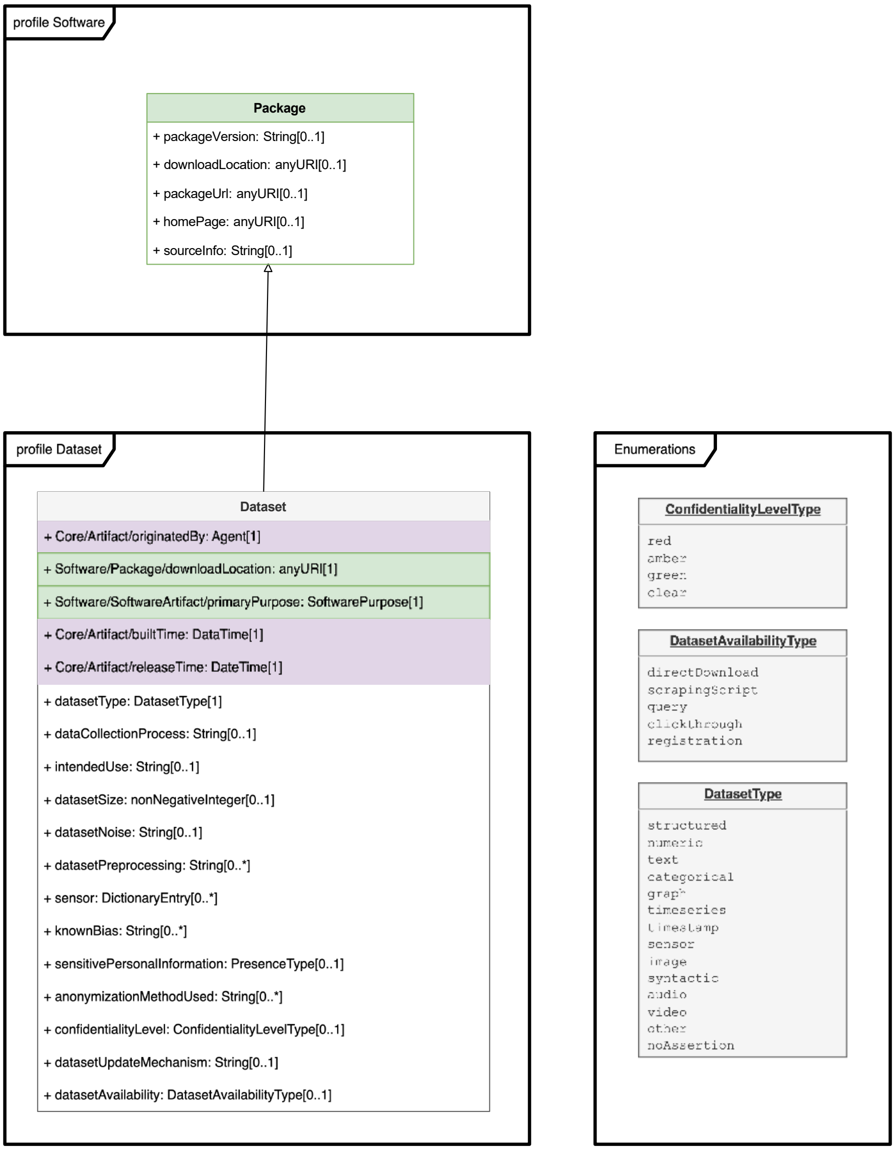
Everything having to do with datasets.

Description

The Dataset profile provides meta-data about data files. Figure 9 below shows the logical model for the Dataset profile with its classes and enumerations.

Metadata

Name	Dataset



profile Software
Package
+ packageVersion: String[0..1]
+ downloadLocation: anyURI[0..1]
+ packageUrl: anyURI[0..1]
+ homePage: anyURI[0..1]
+ sourceInfo: String[0..1]
profile Dataset
Dataset
+ Core/Artifact/originatedBy: Agent[1]
+ Software/Package/downloadLocation: anyURI[1]
+ Software/SoftwareArtifact/primaryPurpose: SoftwarePurpose[1]
+ Core/Artifact/builtTime: DataTime[1]
+ Core/Artifact/releaseTime: DateTime[1]
+ datasetType: DatasetType[1]
+ dataCollectionProcess: String[0..1]
+ intendedUse: String[0..1]
+ datasetSize: nonNegativeInteger[0..1]
+ datasetNoise: String[0..1]
+ datasetPreprocessing: String[0..*]
+ sensor: DictionaryEntry[0..*]
+ knownBias: String[0..*]
+ sensitivePersonalInformation: PresenceType[0..1]
+ anonymizationMethodUsed: String[0..*]
+ confidentialityLevel: ConfidentialityLevelType[0..1]
+ datasetUpdateMechanism: String[0..1]
+ datasetAvailability: DatasetAvailabilityType[0..1]
Enumerations
ConfidentialityLevelType
red
amber
green
clear
DatasetAvailabilityType
directDownload
scrapingScript
query
clickthrough
registration
DatasetType
structured
numeric
text
categorical
graph
timeseries
timestamp
sensor
image
syntactic
audio
video
other
noAssertion

Figure 13 – Dataset Model profile and enumerations

12.1 Dataset Classes

12.1.1 Dataset

Summary

Provides information about the fields in the Dataset profile.

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/v3/Dataset/Dataset
```

Name	Dataset
Instantiability	Concrete
SubclassOf	/Software/Package

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
intendedUse	xsd:string	0	1
knownBias	xsd:string	0	*
sensitivePersonalInformation	/Core/PresenceType	0	1
sensor	/Core/DictionaryEntry	0	*

12.2 Dataset Properties

12.2.1 anonymizationMethodUsed

Summary

Describes the anonymization methods used.

Description

AnonymizationMethodUsed describes the methods used to anonymize the dataset (of fields in the dataset).

Metadata

`https://spdx.org/rdf/v3/Dataset/anonymizationMethodUsed`

Name	anonymizationMethodUsed
Nature	DataProperty
Range	xsd:stringReferenced

- /Dataset/Dataset

12.2.2 confidentialityLevel

Summary

Describes the confidentiality level of the data points contained in the dataset.

Description

ConfidentialityLevel describes the levels of confidentiality of the data points contained in the dataset.

Metadata

`https://spdx.org/rdf/v3/Dataset/confidentialityLevel`

Name	confidentialityLevel
Nature	ObjectProperty
Range	ConfidentialityLevelType

Referenced

- /Dataset/Dataset

12.2.3 dataCollectionProcess

Summary

Describes how the dataset was collected.

Description

DataCollectionProcess describes how a dataset was collected. Examples include the sources from which a dataset was scrapped or the interview protocol that was used for data collection.

Metadata

```
https://spdx.org/rdf/v3/Dataset/dataCollectionProcess
```

Name	dataCollectionProcess
Nature	DataProperty
Range	xsd:string

Referenced

- /Dataset/Dataset

12.2.4 dataPreprocessing

Summary

Describes the preprocessing steps that were applied to the raw data to create the given dataset.

Description

DataPreprocessing describes the various preprocessing steps that were applied to the raw data to create the dataset.

Metadata

```
https://spdx.org/rdf/v3/Dataset/dataPreprocessing
```

Name	dataPreprocessing
Nature	DataProperty
Range	xsd:string

Referenced

- /Dataset/Dataset

12.2.5 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/v3/Dataset/datasetAvailability
```

Name	datasetAvailability
Nature	DataProperty
Range	DatasetAvailabilityType

Referenced

- /Dataset/Dataset

12.2.6 datasetNoise

Summary

Describes potentially noisy elements of the dataset.

Description

DatasetNoise describes what kinds of noises a dataset might encompass. The field uses free form text to specify the fields or the 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/v3/Dataset/datasetNoise
```

Name	datasetNoise
Nature	DataProperty
Range	xsd:string

Referenced

- /Dataset/Dataset

12.2.7 datasetSize

Summary

Captures the size of the dataset.

Description

DatasetSize Captures how large a dataset is. The size is to be measured in bytes.

Metadata

```
https://spdx.org/rdf/v3/Dataset/datasetSize
```

Name	datasetSize

Nature	DataProperty
Range	xsd:nonNegativeInteger

Referenced

- /Dataset/Dataset

12.2.8 datasetType

Summary

Describes the type of the given dataset.

Description

Type 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/v3/Dataset/datasetType`

Name	datasetType
Nature	DataProperty
Range	DatasetType

Referenced

- /Dataset/Dataset

12.2.9 datasetUpdateMechanism

Summary

Describes a mechanism to update the dataset.

Description

DatasetUpdateMechanism describes a mechanism to update the dataset.

Metadata

`https://spdx.org/rdf/v3/Dataset/datasetUpdateMechanism`

Name	datasetUpdateMechanism
Nature	DataProperty
Range	xsd:string

Referenced

- /Dataset/Dataset

12.2.10 intendedUse

Summary

Describes what the given dataset should be used for.

Description

IntendedUse 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/v3/Dataset/intendedUse
```

Name	intendedUse
Nature	DataProperty
Range	xsd:string

Referenced

- /Dataset/Dataset

12.2.11 knownBias

Summary

Records the biases that the dataset is known to encompass.

Description

KnownBias is a free form text field that describes the different biases that the dataset encompasses.

Metadata

```
https://spdx.org/rdf/v3/Dataset/knownBias
```

Name	knownBias
Nature	DataProperty
Range	xsd:string

Referenced

- /Dataset/Dataset

12.2.12 sensitivePersonalInformation

Summary

Describes if any sensitive personal information is present in the dataset.

Description

SensitivePersonalInformation indicates the presence of sensitive personal data or information that allows drawing conclusions about a person's identity.

Metadata

```
https://spdx.org/rdf/v3/Dataset/sensitivePersonalInformation
```

Name	sensitivePersonalInformation
Nature	ObjectProperty
Range	/Core/PresenceType

Referenced

- /Dataset/Dataset

12.2.13 sensor

Summary

Describes a sensor used for collecting the data.

Description

Sensor describes a sensor that was used for collecting the data and its calibration value as a key-value pair.

Metadata

```
https://spdx.org/rdf/v3/Dataset/sensor
```

Name	sensor
Nature	ObjectProperty
Range	/Core/DictionaryEntry

Referenced

- /Dataset/Dataset

12.3 Dataset Vocabularies

12.3.1 ConfidentialityLevelType

Summary

Categories of confidentiality level.

Description

Describes the different confidentiality levels as given by the Traffic Light Protocol.

Metadata

```
https://spdx.org/rdf/v3/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.

12.3.2 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/v3/Dataset/DatasetAvailabilityType
```

Name	DatasetAvailabilityType

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.

12.3.3 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/v3/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.

13 AI Profile

Summary

Additional metadata based on software profile, that is useful for ai applications and models.

Description

The AI profile namespace defines concepts related to AI application and model artifacts. Figure 10 below shows the logical model for the AI profile with its classes and enumerations.

Metadata

Name	AI

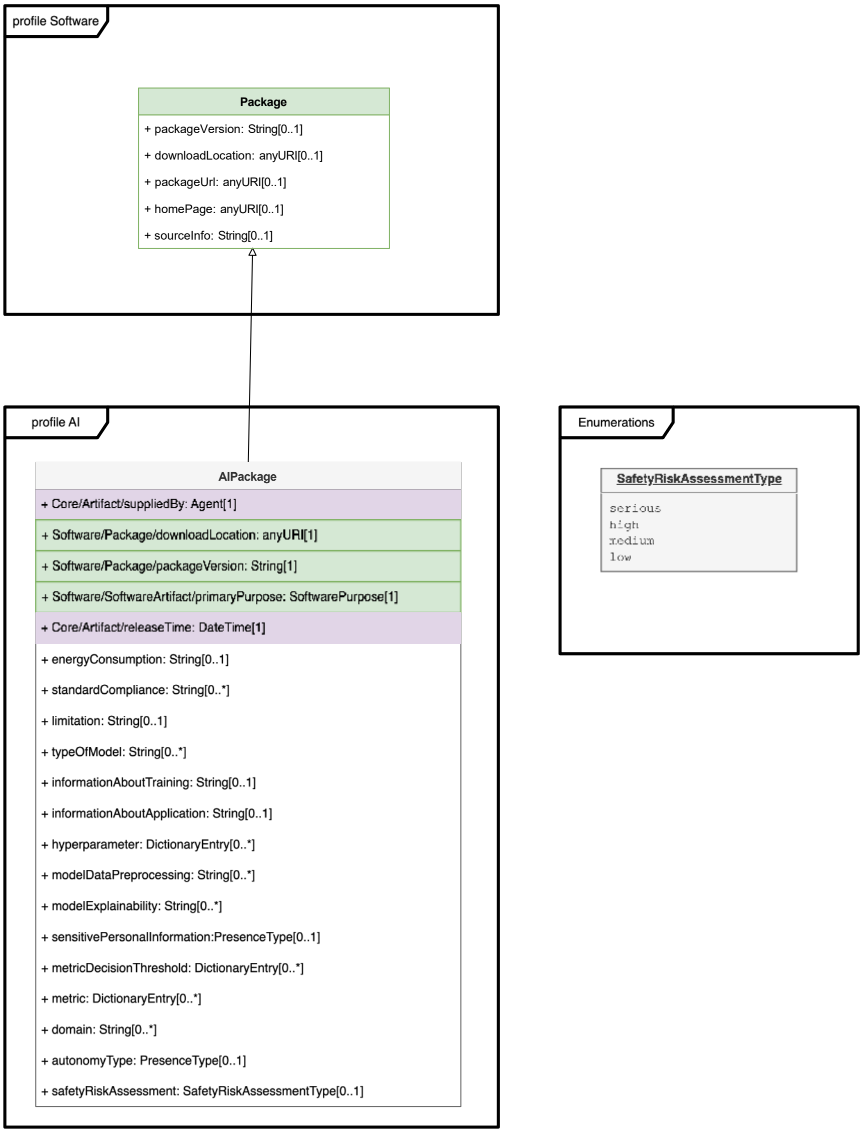

profile Software
Package
+ packageVersion: String[0..1]
+ downloadLocation: anyURI[0..1]
+ packageUrl: anyURI[0..1]
+ homePage: anyURI[0..1]
+ sourceInfo: String[0..1]
profile AI
AIPackage
+ Core/Artifact/suppliedBy: Agent[1]
+ Software/Package/downloadLocation: anyURI[1]
+ Software/Package/packageVersion: String[1]
+ Software/SoftwareArtifact/primaryPurpose: SoftwarePurpose[1]
+ Core/Artifact/releaseTime: DateTime[1]
+ energyConsumption: String[0..1]
+ standardCompliance: String[0..*]
+ limitation: String[0..1]
+ typeOfModel: String[0..*]
+ informationAboutTraining: String[0..1]
+ informationAboutApplication: String[0..1]
+ hyperparameter: DictionaryEntry[0..*]
+ modelDataPreprocessing: String[0..*]
+ modelExplainability: String[0..*]
+ sensitivePersonalInformation:PresenceType[0..1]
+ metricDecisionThreshold: DictionaryEntry[0..*]
+ metric: DictionaryEntry[0..*]
+ domain: String[0..*]
+ autonomyType: PresenceType[0..1]
+ safetyRiskAssessment: SafetyRiskAssessmentType[0..1]
Enumerations
SafetyRiskAssessmentType
serious
high
medium
low

Figure 14 – AI Model profile and enumerations

13.1 AI Classes

13.1.1 AIPackage

Summary

Provides information about the fields in the AI package profile.

Description

Metadata information that can be added to a package to describe an AI application or trained AI model.

Metadata

```
https://spdx.org/rdf/v3/AI/AIPackage
```

Name	AIPackage
Instantiability	Concrete
SubclassOf	/Software/Package

Properties

Property	Type	minCount	maxCount
autonomyType	/Core/PresenceType	0	1
domain	xsd:string	0	*
energyConsumption	xsd:string	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
sensitivePersonalInformation	/Core/PresenceType	0	1
standardCompliance	xsd:string	0	*
typeOfModel	xsd:string	0	*

13.1.2 autonomyType

Summary

States if a human is involved in the decisions of the AI software.

Description

AutonomyType indicates if a human is involved in any of the decisions of the AI software or if that software is fully automatic.

Metadata

```
https://spdx.org/rdf/v3/AI/autonomyType
```

Name	autonomyType
Nature	ObjectProperty
Range	/Core/PresenceType

Referenced

- /AI/AIPackage

13.2 AI Properties

13.2.1 domain

Summary

Captures the domain in which the AI package can be used.

Description

Domain describes the domain in which the AI model contained in the AI software can be expected to operate successfully. Examples include computer vision, natural language etc.

Metadata

```
https://spdx.org/rdf/v3/AI/domain
```

Name	domain
Nature	DataProperty
Range	xsd:string

Referenced

- /AI/AIPackage

13.2.2 energyConsumption

Summary

Indicates the amount of energy consumed to build the AI package.

Description

EnergyConsumption captures the amount of energy needed to train and operate the AI model. This value is also known as training energy consumption or inference energy consumption.

Metadata

```
https://spdx.org/rdf/v3/AI/energyConsumption
```

Name	energyConsumption
Nature	DataProperty
Range	xsd:string

Referenced

- /AI/AIPackage

13.2.3 hyperparameter

Summary

Records a hyperparameter used to build the AI model contained in the AI package.

Description

This field records a hyperparameter value. Hyperparameters are parameters of the machine learning model that are used to control the learning process, for example the optimization and learning rate used during the training of the model.

Metadata

```
https://spdx.org/rdf/v3/AI/hyperparameter
```

Name	hyperparameter
Nature	ObjectProperty
Range	/Core/DictionaryEntry

Referenced

- /AI/AIPackage

13.2.4 informationAboutApplication

Summary

Provides relevant information about the AI software, not including the model description.

Description

InformationAboutApplication describes any relevant information in free form text about how the AI model is used inside the software, as well as any relevant pre-processing steps, third party APIs etc.

Metadata

```
https://spdx.org/rdf/v3/AI/informationAboutApplication
```

Name	informationAboutApplication
Nature	DataProperty
Range	xsd:string

Referenced

- /AI/AIPackage

13.2.5 informationAboutTraining

Summary

Describes relevant information about different steps of the training process.

Description

InformationAboutTraining describes the specific steps involved in the training of the AI model. For example, it can be specified whether supervised fine-tuning or active learning is used as part of training the model.

Metadata

```
https://spdx.org/rdf/v3/AI/informationAboutTraining
```

Name	informationAboutTraining
Nature	DataProperty
Range	xsd:string

Referenced

- /AI/AIPackage

13.2.6 limitation

Summary

Captures a limitation of the AI software.

Description

Limitation captures a limitation of the AI Package (or of the AI models present in the AI package), expressed as free form text. 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/v3/AI/limitation
```

Name	limitation
Nature	DataProperty
Range	xsd:string

Referenced

- /AI/AIPackage

13.2.7 metric

Summary

Records the measurement of prediction quality of the AI model.

Description

Metric 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/v3/AI/metric`

Name	metric
Nature	ObjectProperty
Range	/Core/DictionaryEntry

Referenced

- /AI/AIPackage

13.2.8 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/v3/AI/metricDecisionThreshold`

Name	metricDecisionThreshold
Nature	ObjectProperty
Range	/Core/DictionaryEntry

Referenced

- /AI/AIPackage

13.2.9 modelDataPreprocessing

Summary

Describes all the preprocessing steps applied to the training data before the model training.

Description

ModelDataPreprocessing is 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/v3/AI/modelDataPreprocessing`

Name	modelDataPreprocessing
Nature	DataProperty
Range	xsd:string

Referenced

- /AI/AIPackage

13.2.10 modelExplainability

Summary

Describes methods that can be used to explain the model.

Description

ModelExplainability is a free form text that lists the different explainability mechanisms (such as SHAP, or other model specific explainability mechanisms) that can be used to explain the model.

Metadata

`https://spdx.org/rdf/v3/AI/modelExplainability`

Name	modelExplainability
Nature	DataProperty
Range	xsd:string

Referenced

- /AI/AIPackage

13.2.11 safetyRiskAssessment

Summary

Categorizes safety risk impact of AI software.

Description

SafetyRiskAssessment categorizes the safety risk impact of the AI software in accordance with Article 20 of EC Regulation No 765/2008.

Metadata

```
https://spdx.org/rdf/v3/AI/safetyRiskAssessment
```

Name	safetyRiskAssessment
Nature	ObjectProperty
Range	SafetyRiskAssessmentType

Referenced

- /AI/AIPackage

13.2.12 sensitivePersonalInformation

Summary

Records if sensitive personal information is used during model training.

Description

SensitivePersonalInformation 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.

Metadata

```
https://spdx.org/rdf/v3/AI/sensitivePersonalInformation
```

Name	sensitivePersonalInformation
Nature	ObjectProperty
Range	/Core/PresenceType

Referenced

- /AI/AIPackage

13.2.13 standardCompliance

Summary

Captures a standard that is being complied with.

Description

StandardCompliance captures a standard that the AI software complies with. This includes both published and unpublished standards, for example ISO, IEEE, ETSI etc. The standard could (but not necessarily have to) be used to satisfy a legal or regulatory requirement.

Metadata

```
https://spdx.org/rdf/v3/AI/standardCompliance
```

Name	standardCompliance
Nature	DataProperty
Range	xsd:string

Referenced

- /AI/AIPackage

13.2.14 typeOfModel

Summary

Records the type of the model used in the AI software.

Description

TypeOfModel 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/v3/AI/typeOfModel
```

Name	typeOfModel
Nature	DataProperty
Range	xsd:string

Referenced

- /AI/AIPackage

13.3 AI Vocabularies

13.3.1 SafetyRiskAssessmentType

Summary

Categories of safety risk impact of the application.

Description

Lists the different safety risk type values that can be used to describe the safety risk of AI software according to Article 20 of Regulation 765/2008/EC.

Metadata

`https://spdx.org/rdf/v3/AI/SafetyRiskAssessmentType`

Name	SafetyRiskAssessmentType

Entries

- high: The second-highest level of risk posed by an AI software.
- low: Low/no risk is posed by the AI software.
- medium: The third-highest level of risk posed by an AI software.
- serious: The highest level of risk posed by an AI software.

14 Build Profile

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:

- hasInputs: Describes the relationship from the Build element to its inputs.
- hasOutputs: 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.

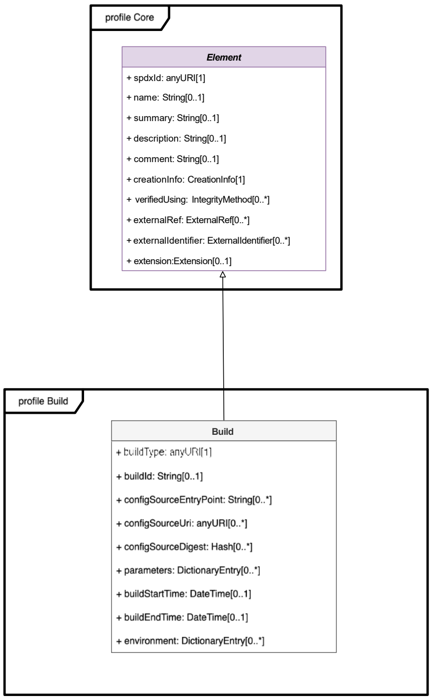
- hasHost: Describes the relationship from the Build element to the build stage or host.
- configures: Describes the relationship from a configuration to the Build element.
- ancestorOf: Describes a relationship from a Build element to Build eelements that describe its child builds.
- decendentOf: Describes 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 hasInputs 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

Name	Build


profile Core
Element
+ spdxId: anyURI[1]
+ name: String[0..1]
+ summary: String[0..1]
+ description: String[0..1]
+ comment: String[0..1]
+ creationInfo: CreationInfo[1]
+ verifiedUsing: IntegrityMethod[0..*]
+ externalRef: ExternalRef[0..*]
+ externalIdentifier: ExternalIdentifier[0..*]
+ extension:Extension[0..1]
profile Build
Build
+ buildType: anyURI[1]
+ buildId: String[0..1]
+ configSourceEntryPoint: String[0..*]
+ configSourceUri: anyURI[0..*]
+ configSourceDigest: Hash[0..*]
+ parameters: DictionaryEntry[0..*]
+ buildStartTime: DateTime[0..1]
+ buildEndTime: DateTime[0..1]
+ environment: DictionaryEntry[0..*]

14.1 Build Classes

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.).

Definitions of "buildType", "configSourceEntrypoint", "configSourceUri", "parameters" and "environment" follow those defined in SLSA provenance.

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/v3/Build/Build
```

Name	Build
Instantiability	Concrete
SubclassOf	/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	*
parameters	/Core/DictionaryEntry	0	*

14.2 Build Properties

14.2.1 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/v3/Build/buildEndTime
```

Name	buildEndTime
Nature	DataProperty
Range	/Core/DateTime

Referenced

- /Build/Build

14.2.2 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. This identifier differs based on build toolchain, platform, or naming convention used by an organization or standard.

Metadata

```
https://spdx.org/rdf/v3/Build/buildId
```

Name	buildId
Nature	DataProperty
Range	xsd:string

Referenced

- /Build/Build

14.2.3 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/v3/Build/buildStartTime`

Name	buildStartTime
Nature	DataProperty
Range	/Core/DateTime

Referenced

- /Build/Build

14.2.4 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 a URI expressing the toolchain, platform, or infrastructure that the build was invoked on. For example, if the build was invoked on GitHub's CI platform using github actions, the buildType can be expressed as https://github.com/actions. In contrast, if the build was invoked on a local machine, the buildType can be expressed as file://username@host/path/to/build.

Metadata

`https://spdx.org/rdf/v3/Build/buildType`

Name	buildType
Nature	DataProperty
Range	xsd:anyURI

Referenced

- /Build/Build

14.2.5 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. This Property uses the Core model's Hash class.

Metadata

```
https://spdx.org/rdf/v3/Build/configSourceDigest
```

Name	configSourceDigest
Nature	ObjectProperty
Range	/Core/Hash

Referenced

- /Build/Build

14.2.6 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. 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:
tags: "*"

jobs:
publish:
runs-on: ubuntu-latest
if: startsWith(github.ref, 'refs/tags/')
steps:

...
```

Metadata

```
https://spdx.org/rdf/v3/Build/configSourceEntrypoint
```

Name	configSourceEntrypoint
Nature	DataProperty
Range	xsd:string

Referenced

- /Build/Build

14.2.7 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. 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. m

Metadata

```
https://spdx.org/rdf/v3/Build/configSourceUri
```

Name	configSourceUri
Nature	DataProperty
Range	xsd:anyURI

Referenced

- /Build/Build

14.2.8 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. This is different from the parameters 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/v3/Build/environment
```

Name	environment
Nature	ObjectProperty
Range	/Core/DictionaryEntry

Referenced

- /Build/Build

14.2.9 parameters

Summary

Property describing the parameters used in an instance of a build.

Description

parameters is a key-value map of all build parameters and their values that were provided to the builder for a build instance. This is different from the environment property in that the keys and values are provided as command line arguments or a configuration file to the builder.

Metadata

`https://spdx.org/rdf/v3/Build/parameters`

Name	parameters
Nature	ObjectProperty
Range	/Core/DictionaryEntry

Referenced

- /Build/Build

15 Lite Profile

Summary

The SPDX Lite profile defines a subset of the SPDX specification, from the point of view of use cases in some industries. SPDX Lite aims at the balance between the SPDX standard and actual workflows in some industries.

Description

The SPDX Lite profile consists of mandatory fields from the Document Creation and Package Information sections and other basic information.

The mandatory part of the Package information in SPDX Lite is basic but useful for complying with licenses. It is easy to understand licensing information by reading an SPDX Lite file. It is easy to create manually an SPDX Lite file by anyone who does not have enough knowledge about licensing information, so that tools are not necessarily required to create an SPDX Lite file.

SPDX Lite has affinity with SPDX tools due to its containing the mandatory part of the Document Creation and Package Information in the SPDX Lite definition.

An SPDX Lite document can be used in parallel with SPDX documents in software supply chains.

Metadata

Name	Lite

16 Extension Profile

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. Figure 12 below shows the logical model for the Extension profile.

Metadata

Name	Extension

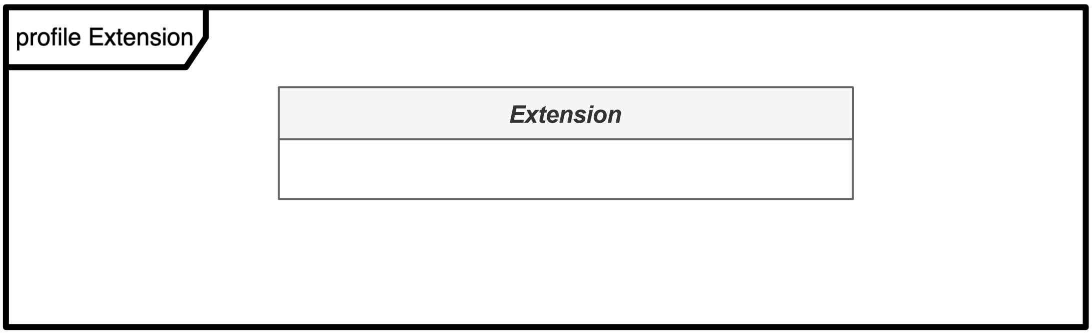


Figure 16 – Extension Model profile

16.1 Extension Classes

16.1.1 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/v3/Extension/Extension`

Name	Extension
Instantiability	Abstract

Properties

Property	Type	minCount	maxCount

This page intentionally left blank.

Annex A: SPDX license expressions

(normative)

A.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.

```
idstring = 1*(ALPHA /DIGIT /"-" /"." )

license-id = <short form license identifier in Annex A.1>

license-exception-id = <short form license exception identifier in Annex A.2>

license-ref = ["DocumentRef-"(idstring)":"]"LicenseRef-"(idstring)

addition-ref = ["DocumentRef-"(idstring)":"]"AdditionRef-"(idstring)

simple-expression = license-id /license-id"+" /license-ref

addition-expression = license-exception-id /addition-ref

compound-expression = (simple-expression /

  simple-expression "WITH" addition-expression /

  compound-expression "AND" compound-expression /

  compound-expression "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.

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.

A.2 Case sensitivity

License expression operators (`AND`, `OR` and `WITH`) should be matched in a *case-sensitive* manner.

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. 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.

A.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.

A.4 Composite license expressions

A.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 D.4.5.

A.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
```

A.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
```

A.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)":"]"AdditonRef-"(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.

A.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.

A.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 B: Using SPDX license list short identifiers in source files

(Informative)

B.1 Introduction

Identifying the license for open source software is critical for both reporting purposes and license compliance. However, determining the license can sometimes be difficult due to a lack of information or ambiguous information. Even when licensing information is present, a lack of consistent notation can make automating the task of license detection very difficult, thus requiring vast amounts of human effort.

Short identifiers from the SPDX License List can be used to indicate license info at the file level. The advantages of doing this are numerous but include:

- It is precise.
- It is concise.
- It is language neutral.
- It is easy and more reliable to machine process.
- Leads to code that is easier to reuse.
- The license information travels with the file (as sometimes not entire projects are used or license files are removed).
- It is a standard and can be universal. There is no need for variation.
- An SPDX short identifier is immutable.
- Easy look-ups and cross-references to the SPDX License List website.

If using SPDX short identifiers in individual files, it is recommended to reproduce the full license in the projects LICENSE file and indicate that SPDX short identifiers are being used to refer to it. For links to projects illustrating these scenarios, see https://spdx.dev/ids-where.

B.2 Format for SPDX-License-Identifier

The SPDX-License-Identifier tag declares the license the file is under and should be placed at or near the top of the file in a comment.

The SPDX License Identifier syntax may consist of a single license (represented by a short identifier from the SPDX license list) or a compound set of licenses (represented by joining together multiple licenses using the license expression syntax).

The tag should appear on its own line in the source file, generally as part of a comment.

```
SPDX-License-Identifier: <SPDX License Expression>
```

B.3 Representing single license

A single license is represented by using the short identifier from SPDX license list, optionally with a unary "+" operator following it to indicate "or later" versions may be applicable.

Examples:

```
SPDX-License-Identifier: CDDL-1.0+
SPDX-License-Identifier: MIT
```

B.4 Representing multiple licenses

Multiple licenses can be represented using an SPDX license expression as defined in Annex D. A set of licenses may optionally be enclosed in parentheses, but are not required to be enclosed. As further described there:

1. When there is a choice between licenses ("disjunctive license"), they should be separated with "OR". If presented with a choice between two or more licenses, use the disjunctive binary "OR" operator to construct a new license expression.
2. Similarly when multiple licenses need to be simultaneously applied ("conjunctive license"), they should be separated with "AND". If required to simultaneously comply with two or more licenses, use the conjunctive binary "AND" operator to construct a new license expression.
3. In some cases, a set of license terms apply except under special circumstances, in this case, use the "WITH" operator followed by one of the recognized exception identifiers.
4. The expression MUST be on a single line, and MUST NOT include a line break in the middle of the expression.

Examples:

```
SPDX-License-Identifier: GPL-2.0-only OR MIT
SPDX-License-Identifier: LGPL-2.1-only AND BSD-2-Clause
SPDX-License-Identifier: GPL-2.0-or-later WITH Bison-exception-2.2
```

Please see Annex D for more examples and details of the license expression specific syntax.

If you can’t express the license(s) as an expression using identifiers from the SPDX list, it is probably best to just put the text of your license header in the file (if there is a standard header), or refer to a neutral site URL where the text can be found. To request a license be added to the SPDX License List, please follow the process described here: https://github.com/spdx/license-list-XML/blob/master/CONTRIBUTING.md.

Alternatively, you can use a `LicenseRef-` custom license identifier to refer to a license that is not on the SPDX License List, such as the following:

```
SPDX-License-Identifier:  LicenseRef-my-special-license
```

The `LicenseRef-` format is defined in Annex D. When using a custom `LicenseRef-` identifier, you will also need to provide a way for others to determine what license text corresponds to it. Version 3.0 of the REUSE Software Specification provides a standardized format that can optionally be used for providing the corresponding license text for these identifiers.

Annex C: References

(Informative)

[1] 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

[2] Dan Geer and Joshua Corman, “Almost Too Big to Fail”, Usenix ;login article, Vol. 39. No. 4, August 2014, https://www.usenix.org/system/files/login/articles/15_geer_0.pdf

[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, chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.govinfo.gov/content/pkg/CHRG-115hhrg27760/pdf/CHRG-115hhrg27760.pdf

[4] CISQ Software Bill of Materials project, “Tool-to-Tool Software Bill of Materials Exchange”, https://www.it-cisq.org/software-bill-of-materials/

[5] MITRE, “Standardizing SBOM within the SW Development Tooling Ecosystem”, Nov 2019, https://www.mitre.org/sites/default/files/2021-10/pr-19-01876-16-standardizing-sbom-within-the-sw-development-tooling-ecosystem.pdf

[6] NTIA Software Bill Of Materials web page, https://ntia.gov/sbom/

[7] 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/sites/default/files/2021-11/prs-21-0278-deliver-uncompromised-securing-critical-software-supply-chain.pdf

[8] 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/

[9] 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.doc.gov/files/ntia/publications/sbom_minimum_elements_report.pdf