fix pagination

Instantiability	Abstract

Properties

Property	Type	minCount	maxCount
comment	xsd:string	0	1

Add new All Properties section with details of all inherited properties from classess and super-classes

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

Add new All Properties section with details of all inherited properties from classess and super-classes

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)~~

1 clause

spdx-3-model repository 2

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

1) An SPDX content producer wishing to provide clarity of their serialization of an SPDX 2.X simple style collection where all content is newly minted and a single prefix- namespace is used. The consumer of SPDX content wishes to preserve the name space mapping provided by such a producer. In this case, the consumer would record the namespace map prefixes in the NamespaceMap such that subsequent serializations could reproduce the prefixes /namespaces in the native serialization format.

1../../../serializations.md
2https://github.com/spdx/spdx-3-model/tree/main/serialization

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 Pa c ka ge Ve r i f i c a t i on Code .

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.

RFC 3986 Uniform Resource Identifier (URI): Generic Syntax 3

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

3https://www.rfc-editor.org/info/rfc3986

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/externalIdentifier`

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 pr

Description

This field points to a resource outside the scope of the SPDX-3.0 con Element.

Entries

cpe22 Common Platform Enumeration Specification 2.24

cpe23 Common Platform Enumeration: Naming Specification Version 2.35

cve Common Vulnerabilities and Exposures identifiers, an identifier for a specific software flaw defined within the official CVE Dictionary and that conforms to the CVE specification6.

email Email address, as defined in RFC 36967 Section 3.

gitoid Gitoid8, stands for Git Object ID9. A gitoid of type blob is a unique hash of a binary artifact. A gitoid may represent either an Artifact Identifier10 for the software artifact or an Input Manifest Identifier11 for the software artifact's associated Artifact Input Manifest12; this ambiguity exists because the Artifact Input Manifest is itself an artifact, and the gitoid of that artifact is its valid identifier. Gitoids calculated on soft- ware artifacts (Snippet, File, or Package Elements) should be recorded in the SPDX 3.0 SoftwareArtifact's contentIdentifier property. Gitoids calculated on the Artifact Input Manifest (Input Manifest Identifier) should be recorded in the SPDX 3.0 Element's externalIdentifier property. See OmniBOR Specification13,a minimalistic specification for describing software Artifact Dependency Graphs14.

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

packageUrl Package URL, as defined in the corresponding Annex15 of this specification. **securityOther** Used when there is a security related identifier of unspecified type.

swhid SoftWare Hash IDentifier, a persistent intrinsic identifier for digital artifacts, such as files, trees (also known as directories or folders), commits, and other objects typically found in version control systems. The format of the identifiers is defined in the SWHID specification16 (ISO/IEC DIS 18670). They typically look like `swh:1:cnt:94a9ed024d3859793618152ea559a168bbcbb5e2`.

4https://cpe.mitre.org/files/cpe-specification_2.2.pdf
5https://csrc.nist.gov/publications/detail/nistir/7695/final
6https://csrc.nist.gov/glossary/term/cve_id
7https://www.rfc-editor.org/info/rfc3986
8https://www.iana.org/assignments/uri-schemes/prov/gitoid
9https://git-scm.com/book/en/v2/Git-Internals-Git-Objects
10https://github.com/omnibor/spec/blob/eb1ee5c961c16215eb8709b2975d193a2007a35d/spec/SPEC.md#artifact-identifier-t ypes
11https://github.com/omnibor/spec/blob/eb1ee5c961c16215eb8709b2975d193a2007a35d/spec/SPEC.md#input-manifest-ide ntifier
12https://github.com/omnibor/spec/blob/eb1ee5c961c16215eb8709b2975d193a2007a35d/spec/SPEC.md#artifact-input-manif est
13https://github.com/omnibor/spec/
14https://github.com/omnibor/spec/blob/eb1ee5c961c16215eb8709b2975d193a2007a35d/spec/SPEC.md#artifact-dependenc
15../../../annexes/pkg-url-specification.md 16https://www.swhid.org/specification/v1.1/4.Syntax

Metadata

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

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 ext

Metadata

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

Name	externalRefType
Nature	ObjectProperty
Range	ExternalRefType

Referenced

- /Core/ExternalRef

8.2.24 externalSpdxId

Summary

Identifies an external Element used within a Doc

Description

ExternalSpdxId identifies an external Element us

Metadata

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

Name	externalSpdxId
Nature	DataProperty
Range	xsd:anyURI

Referenced

- /Core/ExternalMap

Entries

altDownloadLocation A reference to an alternative download location.
altWebPage A reference to an alternative web page.
binaryArtifact A reference to binary artifacts related to a package.
bower A reference to a Bower package. The package locator format, looks like `package#version`, is defined in the "install" section of Bower API documentation19.
buildMeta A reference build metadata related to a published package.
buildSystem A reference build system used to create or publish the package.
certificationReport A reference to a certification report for a package from an accredited/independent body.
chat A reference to the instant messaging system used by the maintainer for a package.
componentAnalysisReport A reference to a Software Composition Analysis (SCA) report.
cwe Common Weakness Enumeration20. A reference to a source of software flaw defined within the official CWE List21 that conforms to the CWE specification22.
documentation A reference to the documentation for a package.
dynamicAnalysisReport A reference to a dynamic analysis report for a package.
eolNotice A reference to the End Of Sale (EOS) and/or End Of Life (EOL) information related to a package.
exportControlAssessment A reference to a export control assessment for a package.
funding A reference to funding information related to a package.
issueTracker A reference to the issue tracker for a package.
license A reference to additional license information related to an artifact.
mailingList A reference to the mailing list used by the maintainer for a package.
mavenCentral A reference to a Maven repository artifact. The artifact locator format is defined in the Maven documentation23 and looks like `groupId:artifactId[:version]`.
metrics A reference to metrics related to package such as OpenSSF scorecards.
npm A reference to an npm package. The package locator format is defined in the npm documentation24 and looks like `package@version`.
nuget A reference to a NuGet package. The package locator format is defined in the NuGet documentation25 and looks like `package/version`.
other Used when the type does not match any of the other options.
privacyAssessment A reference to a privacy assessment for a package.
productMetadata A reference to additional product metadata such as reference within organization's product catalog.
purchaseOrder A reference to a purchase order for a package. **qualityAssessmentReport** A reference to a quality assessment for a package. **releaseHistory** A reference to a published list of releases for a package. **releaseNotes** A reference to the release notes for a package.
riskAssessment A reference to a risk assessment for a package. **runtimeAnalysisReport** A reference to a runtime analysis report for a package.
secureSoftwareAttestation A reference to information assuring that the software is developed using security practices as defined by NIST SP 800-218 Secure Software Development Framework (SSDF) Version 1.126 or CISA Secure Software Development Attestation Form27.
securityAdversaryModel A reference to the security adversary model for a package. **securityAdvisory** A reference to a published security advisory (where advisory as defined per ISO 29147:201828) that may affect one or more elements, e.g., vendor advisories or specific NVD entries. **securityFix** A reference to the patch or source code that fixes a vulnerability.
securityOther A reference to related security information of unspecified type.
securityPenTestReport A reference to a penetration test29 report for a package.
securityPolicy A reference to instructions for reporting newly discovered security vulnerabilities for a package.
securityThreatModel A reference the security threat model30 for a package.
socialMedia A reference to a social media channel for a package.
sourceArtifact A reference to an artifact containing the sources for a package. **staticAnalysisReport** A reference to a static analysis report for a package.
support A reference to the software support channel or other support information for a package. **vcs** A reference to a version control system related to a software artifact.
vulnerabilityDisclosureReport AreferencetoaVulnerabilityDisclosureReport(VDR)whichprovidesthesoft- ware supplier's analysis and findings describing the impact (or lack of impact) that reported vulnerabilities have on packages or products in the supplier's SBOM as defined in NIST SP 800-161 Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations31.
vulnerabilityExploitabilityAssessment A reference to a Vulnerability Exploitability eXchange (VEX) state- ment which provides information on whether a product is impacted by a specific vulnerability in an included package and, if affected, whether there are actions recommended to remediate. See also NTIA VEX one-page summary32.

17https://www.rfc-editor.org/info/rfc9393
18https://www.iana.org/assignments/uri-schemes/uri-schemes.xhtml
19https://bower.io/docs/api/#install
20https://csrc.nist.gov/glossary/term/common_weakness_enumeration
21https://cwe.mitre.org/data/
22https://cwe.mitre.org/
23https://maven.apache.org/guides/mini/guide-naming-conventions.html
24https://docs.npmjs.com/cli/v10/configuring-npm/package-json
25https://docs.nuget.org
26https://csrc.nist.gov/pubs/sp/800/218/final
27https://www.cisa.gov/resources-tools/resources/secure-software-development-attestation-form
28https://www.iso.org/standard/72311.html
29https://en.wikipedia.org/wiki/Penetration_test 30https://en.wikipedia.org/wiki/Threat_model
31https://csrc.nist.gov/pubs/sp/800/161/r1/final
32https://ntia.gov/files/ntia/publications/vex_one-page_summary.pdf
33https://www.rfc-editor.org/info/rfc1950
34https://www.rfc-editor.org/info/rfc7693
35https://www.rfc-editor.org/info/rfc7693
36https://www.rfc-editor.org/info/rfc7693
37https://github.com/BLAKE3-team/BLAKE3-specs/blob/master/blake3.pdf
38https://pq-crystals.org/dilithium/
39https://pq-crystals.org/kyber/

adler32 Adler-32 checksum is part of the widely used zlib compression library as defined in RFC 195033 Section 2.3.

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

sha224 SHA-2 with a digest length of 224, as defined in RFC 387446.
sha256 SHA-2 with a digest length of 256, as defined in RFC 623447.
sha384 SHA-2 with a digest length of 384, as defined in RFC 623448.
sha3_224 SHA-3 with a digest length of 224, as defined in FIPS 20249.
sha3_256 SHA-3 with a digest length of 256, as defined in FIPS 20250.
sha3_384 SHA-3 with a digest length of 384, as defined in FIPS 20251.
sha3_512 SHA-3 with a digest length of 512, as defined in FIPS 20252.
sha512 SHA-2 with a digest length of 512, as defined in RFC 623453.

Summary

Provide an enumerated set of software lifecycle phases that can provide c

Description

This enumeration summarizes common phases when dependency and ot
based on their context. For example, a build dependency, may have diffe

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`

31https://csrc.nist.gov/pubs/sp/800/161/r1/final
32https://ntia.gov/files/ntia/publications/vex_one-page_summary.pdf
33https://www.rfc-editor.org/info/rfc1950
34https://www.rfc-editor.org/info/rfc7693
35https://www.rfc-editor.org/info/rfc7693
36https://www.rfc-editor.org/info/rfc7693
37https://github.com/BLAKE3-team/BLAKE3-specs/blob/master/blake3.pdf
38https://pq-crystals.org/dilithium/
39https://pq-crystals.org/kyber/
40https://falcon-sign.info/falcon.pdf
41https://www.rfc-editor.org/info/rfc1319/
42https://www.rfc-editor.org/info/rfc1186
43https://www.rfc-editor.org/info/rfc1321
44https://people.csail.mit.edu/rivest/pubs/RABCx08.pdf
45https://www.rfc-editor.org/info/rfc3174
46https://www.rfc-editor.org/info/rfc3874
47https://www.rfc-editor.org/info/rfc6234
48https://www.rfc-editor.org/info/rfc6234
49https://csrc.nist.gov/pubs/fips/202/final
50https://csrc.nist.gov/pubs/fips/202/final
51https://csrc.nist.gov/pubs/fips/202/final
52https://csrc.nist.gov/pubs/fips/202/final
53https://www.rfc-editor.org/info/rfc6234

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

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

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

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.

RFC 2046 MIME Part Two: Media Types 54

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

Examples

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

A list of all possible media types is available at IANA Protocol Registries55.

8.4.3 SemVer

Summary

A string constrained to the SemVer 2.0.0 specification.

Description

56

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-] +) *)) ? $

54https://www.rfc-editor.org/info/rfc2046
55https://www.iana.org/assignments/media-types/media-types.xhtml
56https://semver.org/

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.[57] 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.[58]

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.

57../../../annexes/pkg-url-specification.md
58https://github.com/package-url/purl-spec/blob/b33dda1cf4515efa8eabbbe8e9b140950805f845/PURL-TYPES.rst

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.[59]

Metadata

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

Name	sbomType

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

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., execut
- containers, and virtual machine images) after its build. Such analysi some contexts, this may also be referred to as a "3rd party" SBOM.
- build: SBOM generated as part of the process of building the softwa executable or package) from data such as source files,
- dependencies, built components, build process ephemeral data, and c
- deployed: SBOM provides an inventory of software that is present c other SBOMs that combines analysis of configuration options, and
- examination of execution behavior in a (potentially simulated) deplo
- design: SBOM of intended, planned software project or product witl not yet exist) for a new software artifact.
- runtime: SBOM generated through instrumenting the system runnin present in the system, as well as external call-outs or dynamically lo

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. So Element is being used rather than the content of the Element. This field is a r of the Element from the producer and consumer perspective from which both context in which the Element exists.

Metadata

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

Name	SoftwarePurpose

8.3.1 ContentIdentifierType Summary
Specifies the type of a content identifier.
Description
ContentIdentifierType specifies the type of a content identifier.
Metadata
`https://spdx.org/rdf/3.0.1/terms/Software/ContentIdentifierType`
Name: ContentIdentifierType

gitoid Gitoid60, stands for Git Object ID61. A gitoid of type blob is a unique hash of a binary artifact. A gitoid may represent either an Artifact Identifier62 for the software artifact or an Input Manifest Identifier63 for thesoftwareartifact'sassociatedArtifactInputManifest64;thisambiguityexistsbecausetheArtifactInput Manifest is itself an artifact, and the gitoid of that artifact is its valid identifier. Gitoids calculated on soft- ware artifacts (Snippet, File, or Package Elements) should be recorded in the SPDX 3.0 SoftwareArtifact's contentIdentifier property. Gitoids calculated on the Artifact Input Manifest (Input Manifest Identifier) should be recorded in the SPDX 3.0 Element's externalIdentifier property. See OmniBOR Specification65,a minimalistic specification for describing software Artifact Dependency Graphs66.
swhid SoftWare Hash IDentifier, a persistent intrinsic identifier for digital artifacts, such as files, trees (also known as directories or folders), commits, and other objects typically found in version control systems. The format of the identifiers is defined in the SWHID specification67 (ISO/IEC DIS 18670). They typically look like `swh:1:cnt:94a9ed024d3859793618152ea559a168bbcbb5e2`.

60https://www.iana.org/assignments/uri-schemes/prov/gitoid
61https://git-scm.com/book/en/v2/Git-Internals-Git-Objects
62https://github.com/omnibor/spec/blob/eb1ee5c961c16215eb8709b2975d193a2007a35d/spec/SPEC.md#artifact-identifier-types
63https://github.com/omnibor/spec/blob/eb1ee5c961c16215eb8709b2975d193a2007a35d/spec/SPEC.md#input-manifest-identifier
64https://github.com/omnibor/spec/blob/eb1ee5c961c16215eb8709b2975d193a2007a35d/spec/SPEC.md#artifact-input-manifest
65https://github.com/omnibor/spec/
66https://github.com/omnibor/spec/blob/eb1ee5c961c16215eb8709b2975d193a2007a35d/spec/SPEC.md#artifact-dependency-graph-adg
67https://www.swhid.org/specification/v1.1/4.Syntax

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[68], 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

68https://www.cisa.gov/sites/default/files/2023-04/sbom-types-document-508c.pdf

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.

in A Complete Guide to the Common Vulnerability Scoring System Version 2.0 69.

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

69https://www.first.org/cvss/v2/guide

Properties

Property	Type	minCount	maxCount
score	xsd:decimal	1	1
vectorString	xsd:string	1	1

Add new All Properties section with details of all inherited properties from classess and super-classes

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.

as defined in Common Vulnerability Scoring System v3.0: Specification Document 70 or Common Vulnerability Scoring System v3.1: SpecificationDocument 71.

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"
}
```


70https://www.first.org/cvss/v3.0/specification-document
71https://www.first.org/cvss/v3.1/specification-document

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

Add new All Properties section with details of all inherited properties from classess and super-classes

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.

in Common Vulnerability Scoring System version 4.0: Specification Document 72.

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"
   },
 ],
```


72https://www.first.org/cvss/v4.0/specification-document

```
 "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

Add new All Properties section with details of all inherited properties from classess and super-classes

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.

The EPSS Model 73.

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.

, and the percentile ranking of probability relative to all other vulnerabilities' EPSS scores,

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`

73https://www.first.org/epss/model

fix pagination

SubclassOf	VulnAssessmentRelationship

Properties

Property	Type	minCount	maxCount
percentile	xsd:decimal	1	1
probability	xsd:decimal	1	1
publishedTime	/Core/DateTime	1	1

Add new All Properties section with details of all inherited properties from classess and super-classes

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.

74.

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

Add new All Properties section with details of all inherited properties from classess and super-classes

10.1.6 SsvcVulnAssessmentRelationship

Summary

Provides an SSVC assessment for a vulnerability.

74https://www.cisa.gov/known-exploited-vulnerabilities-catalog

Description

by CISAStakeholder-SpecificVulnerabilityCategorizationGuide 75.

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

Add new All Properties section with details of all inherited properties from classess and super-classes

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",
```


75https://www.cisa.gov/stakeholder-specific-vulnerability-categorization-ssvc

- /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~~ probability score. The definition follows “percentile” in EPSS Data 76.

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~~ The definition follows “epss” in EPSS Data 77.

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.

76https://www.first.org/epss/data_stats
77https://www.first.org/epss/data_stats

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.~~ by Forum of Incident Response and Security Teams 78.

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

78https://www.first.org/cvss/

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

Common Vulnerability Scoring System Vector String 79.

Metadata

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

Name	vectorString
Nature	DataProperty
Range	xsd:string

Referenced

79https://www.first.org/cvss/v4.0/specification-document#Vector-String

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.

Common Vulnerability Scoring System v3.0: Specification Document 80 and Common Vulnerability Scoring System version 4.0: Specification Document 81. CvssSeverityType is a mandatory field because baseSeverity is required in the CVSS 3.0 schema 82, CVSS 3.1 schema83, and CVSS 4.0 schema 84.

80https://www.first.org/cvss/v3.0/specification-document#Qualitative-Severity-Rating-Scale
81https://www.first.org/cvss/v4.0/specification-document#Qualitative-Severity-Rating-Scale
82https://www.first.org/cvss/cvss-v3.0.json
83https://www.first.org/cvss/cvss-v3.1.json
84https://www.first.org/cvss/cvss-v4.0.json

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 85.

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.

85https://www.cisa.gov/stakeholder-specific-vulnerability-categorization-ssvc

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[87] 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.[88]

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

Add new All Properties section with details of all inherited properties from classess and super-classes

87../../annexes/spdx-license-expressions.md
88https://spdx.org/licenses/

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

Add new All Properties section with details of all inherited properties from classess and super-classes

AnyLicenseInfo is an abstract class representing a license combination consisting of one or more licenses (op- tionally including additional text), which may be combined according to the SPDX license expression syntax89.
An AnyLicenseInfo is used by licensing properties of software artifacts. It can be:
• a NoneLicense;
• a NoAssertionLicense;
• a single license (either on the SPDX License List90 or a custom-defined license91);
• 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.

89../../../annexes/spdx-license-expressions.md
90https://spdx.org/licenses/
91../../ExpandedLicensing/Classes/CustomLicense.md

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

A LicenseExpression enables the representation, in a single string, of a combination of one or more licenses, together with additions such as license exceptions.
The syntax for a LicenseExpression string is set forth in the corresponding Annex of this specification (“SPDX license expressions” 92). A LicenseExpression string is not valid if it does not conform to the grammar set forth in that annex.
The ExpandedLicensing profile can be used to represent the complete parsed license expression as a combination of license objects.

Add new All Properties section with details of all inherited properties from classes

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`

92. ./../../annexes/spdx-license-expressions.md

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

Add new All Properties section with details of all inherited properties from classess and super-classes

11.1.3 SimpleLicensingText

Summary

A license or addition that is not listed on the SPDX License List.

Description

93

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`

93https://spdx.org/licenses

fix pagination

Instantiability	Concrete
SubclassOf	/Core/Element

Properties

Property	Type	minCount	maxCount
licenseText	xsd:string	1	1

Add new All Properties section with details of all inherited properties from classess and super-classes

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 corresponding CustomLicense, CustomLicenseAddition, or SimpleLicensingText.~~

Metadata

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

Name	customIdToUri
Nature	ObjectProperty
Range	/Core/DictionaryEntry

Referenced

- /SimpleLicensing/LicenseExpression

The License Expression syntax 94 dictates any reference starting with a “LicenseRef-” or “AdditionRef-” refers to license or addition text not found in the official SPDX License List 95.
These custom licenses must be a CustomLicense, a CustomLicenseAddition, or a SimpleLicensingText which are identified with a unique URI identifier.
The key for the DictionaryEntry is the string used in the license expression and the value is the URI for the corresponding CustomLicense, CustomLicenseAddition, or SimpleLicensingText.

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

94../../../annexes/spdx-license-expressions.md
95https://spdx.org/licenses/
92../../../annexes/spdx-license-expressions.md

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

97

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

97https://spdx.org/licenses/

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[98] 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[99] 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

98. ./../../annexes/license-matching-guidelines-and-templates.md
99. ./../annexes/spdx-license-expressions.md

100

"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	*

Add new All Properties section with details of all inherited properties from classess and super-classes

11.2.2 CustomLicense

Summary

A license that is not listed on the SPDX License List.

Description

101

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

Add new All Properties section with details of all inherited properties from classess and super-classes

11.2.3 CustomLicenseAddition

Summary

A license addition that is not listed on the SPDX Exceptions List.

Description

102

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

100../../../annexes/spdx-license-expressions.md
101https://spdx.org/licenses
102https://spdx.org/licenses/exceptions-index.html

the value.

Metadata

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

Name	ExtendableLicense
Instantiability	Abstract
SubclassOf	/SimpleLicensing/AnyLicenseInfo

Properties

Property	Type	minCount	maxCount

Add new All Properties section with details of all inherited properties from classess and super-classes

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

Add new All Properties section with details of all inherited properties from classess and super-classes

11.2.7 License

Summary

Abstract class for the portion of an AnyLicenseInfo representing a license.

Description

103

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`

103https://spdx.org/licenses/

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

Add new All Properties section with details of all inherited properties from classess and super-classes

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.

104

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

Add new All Properties section with details of all inherited properties from classess and super-classes

11.2.9 ListedLicense

Summary

104https://spdx.org/licenses/exceptions-index.html

A license that is listed on the SPDX License List.

Description

105

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

Add new All Properties section with details of all inherited properties from classess and super-classes

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~~.

106

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

Add new All Properties section with details of all inherited properties from classess and super-classes

11.2.11 OrLaterOperator

Summary

105https://spdx.org/licenses
106https://spdx.org/licenses/exceptions-index.html

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

Add new All Properties section with details of all inherited properties from classess and super-classes

11.2.12 WithAdditionOperator

Summary

Portion of an AnyLicenseInfo representing a License which has additional text applied to it.

Description

107

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

Add new All Properties section with details of all inherited properties from classess and super-classes

107https://spdx.org/licenses/exceptions-index.html

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.

License List Matching Guidelines 108

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

A deprecatedVersion, for a ListedLicense on the SPDX License List 109 or a ListedLicenseException on the SPDX License Exceptions 110, specifies which version release of the License List was the first one in which it was marked as deprecated.

108../../../annexes/license-matching-guidelines-and-templates.md
109https://spdx.org/licenses/
110https://spdx.org/licenses/exceptions-index.html

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[111], 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[112], 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.

111https://spdx.org/licenses/exceptions-index.html
112https://spdx.org/licenses/

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

(FSF)113

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~~.

Various Licenses and Comments about Them 114.

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

113https://fsf.org
114https://www.gnu.org/licenses/license-list.en.html

115

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/.

OSI Approved Licenses 116.

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

The formal schema definition is available at SPDX License List XML Schema 117.
For a text description of the XML fields, see XML template fields 118.

115https://opensource.org
116https://opensource.org/licenses
117https://github.com/spdx/license-list-XML/blob/v3.24.0/schema/ListedLicense.xsd
118https://github.com/spdx/license-list-XML/blob/v3.24.0/DOCS/xml-fields.md

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[119] 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

119https://spdx.org/licenses/

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.

120

It is recommended to use licenseXml 121 instead, as it can capture all the text and metadata associated with a license.

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

120../../../annexes/license-matching-guidelines-and-templates.md
121./licenseXml.md

Referenced

- /ExpandedLicensing/License

11.2.26 standardLicenseTemplate

Summary

Identifies the full text of a License, in SPDX templating format.

Description

122

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.

It is recommended to use licenseXml 123 instead, as it can capture all the text and metadata associated with a license.

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

122../../../annexes/license-matching-guidelines-and-templates.md
123./licenseXml.md

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.[124]

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.

124https://en.wikipedia.org/wiki/Traffic_Light_Protocol

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 results from the AI model.

~~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.~~

A free-form text that lists the different explainability mechanisms and how they can be used to explain the results from the AI model.
The mechanisms can be model-agnostic methods, such as SHapley Additive exPlanations (SHAP) 125 and Local Interpretable Model-agnostic Explanations (LIME) 126, and model-specific methods that applied to a limited category of models.

Metadata

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

Name	modelExplainability
Nature	DataProperty
Range	xsd:string

Referenced

- /AI/AIPackage

125https://shap.readthedocs.io/
126https://github.com/marcotcr/lime

13.2.11 safetyRiskAssessment

Summary

Records the results of general safety risk assessment of the AI system.

~~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.~~

Records the results of general safety risk assessment of the AI system. Using categorization according to the EU general risk assessment methodology 127. The methodology implements Article 20 of Regulation (EC) No 765/2008 and is intended to assist authorities when they assess general product safety compliance.

It is important to note that this categorization differs from the one proposed in the EU AI Act's provisional agreement.

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

127https://ec.europa.eu/docsroom/documents/17107

13.3 AI Vocabularies

13.3.1 SafetyRiskAssessmentType

Summary

Specifies the safety risk level.

~~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.~~

Lists the different general safety risk levels that can be used to describe the general safety risk of an AI system. Using categorization according to the EU general risk assessment methodology 128. The methodology implements Article 20 of Regulation (EC) No 765/2008 and is intended to assist authorities when they assess general product safety compliance.

Metadata

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

Name	SafetyRiskAssessmentType

Entries

an

- 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~~.

system

128https://ec.europa.eu/docsroom/documents/17107

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~~. SLSA Provenance v0.21 29.

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	*

Add new All Properties section with details of all inherited properties from classess and super-classes

129https://slsa.dev/provenance/v0.2

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[130]

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

130../../Core/Classes/Hash.md

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.

parameter131

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

131parameter.md

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.~~

parameter is a key-value of a build parameter and its value that was provided to the builder for a build instance. This is different from the environment132 property in that the key and value are provided as command line arguments or a configuration file to the builder.

Metadata

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

Name	parameters
Nature	ObjectProperty
Range	/Core/DictionaryEntry

Referenced

- /Build/Build

132environment.md

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 /"-" /"." )
                                        from SPDX License List
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.

1http://tools.ietf.org/html/rfc5234
2http://tools.ietf.org/html/rfc7405

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

license-id = <short form license identifier from SPDX License List>

license-exception-id = <short form license exception identifier from SPDX License List>

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

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

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

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

compound-expression = (simple-expression /

  simple-expression ( %s"WITH" / %s"with" ) addition-expression /

  compound-expression ( %s"AND" / %s"and" ) compound-expression /

  compound-expression ( %s"OR" / %s"or" ) compound-expression /

  "(" compound-expression ")" )
```

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.[4]

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: "(...)".

3https://spdx.org/licenses
4https://spdx.org/licenses

License expression operators (AND, and, OR, or, WITH and with) should be matched in a case-sensitive manner, i.e., letters must be all upper case or all lower case.

License identifiers (including license exception identifiers) used in SPDX documents or source code files should be matched in a case-insensitive manner. In other words, MIT, Mit and mIt should all be treated as the same identifier and referring to the same license.

However, please be aware that it is often important to match with the case of the canonical identifier on the SPDX License List[3]. This is because the canonical identifier's case is used in the URL of the license's or exception's entry on the List, and because the canonical identifier is translated to a URI in RDF documents.

For user defined license identifiers, only the variable part (after LicenseRef-) is case insensitive. This means, for example, that LicenseRef-Name and LicenseRef-name should be treated as the same identifier and considered to refer to the same license, while licenseref-name is not a valid license identifier.

The same applies to AdditionRef- user defined identifiers.

The current set of valid license exceptions identifiers can be found in spdx.org/licenses. [5]

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

5https://spdx.org/licenses