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

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

Introduction

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

1 Scope

This System Package Data Exchange (SPDX®) specification defines a standard capable of representing systems with software components in as SBOMs (Software Bill of Materials) and other AI, data and security references supporting a range of risk management use cases. An SPDX document can be associated with a set of software packages, files or snippets and contains information about the software in the SPDX format described in this specification.

2 Normative references

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

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

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

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

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

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

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

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

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

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

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

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

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

3 Terms and definitions

To do: Need to fill this in to align to the terms and definitions that apply in this revision of the specification.

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

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

4 Model and serializations

4.1 Overview

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

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

4.2 RDF Serialization

Since the data model is based on RDF, any SPDX data can be serialized in any of the multiple RDF serialization formats, including but not limited to: - JSON-LD format as defined in JSON-LD 1.1; - Turtle (Terse RDF Triple Language) format as defined in RDF 1.1 Turtle; - N-Triples format as defined in RDF 1.1 N-Triples; and - RDF/XML format as defined in RDF 1.1 XML Syntax.

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

4.3 Canonical serialization

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

Such a canonical form normalizes things like ordering and formatting.

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

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

- no line breaks
- key names MUST be wrapped in double quotes

Bibliography

The following documents are useful references for implementers and users of this document:

[1] *Software Package Data Exchange (SPDX®) Specification Version 1.0* and 1.1, 1.2, 2.0, 2.1, 2.2, and 2.3; SPDX.dev, https://spdx.dev/specifications

[2] *Open Source Initiative (OSI)*; https://opensource.org/licenses

Core

Summary

The basis for all SPDX profiles.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core`

Name	Core

Agent

Summary

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

Description

The Agent class represents anything that has the potential to act on a system. This could be a person, organization, software agent, etc. This is not to be confused with tools that are used to perform tasks.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/Agent`

Name	Agent
Instantiability	Concrete
SubclassOf	Element

Properties

Property	Type	minCount	maxCount

Annotation

Summary

An assertion made in relation to one or more elements.

Description

An Annotation is an assertion made in relation to one or more elements. The `contentType` property describes the format of the `statement` property.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/Annotation`

Name	Annotation
Instantiability	Concrete
SubclassOf	Element

Properties

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

Artifact

Summary

A distinct article or unit within the digital domain.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/Artifact`

Name	Artifact
Instantiability	Abstract
SubclassOf	Element

Properties

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

Bom

Summary

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

Description

A Bill Of Materials (BOM) is a container for a grouping of SPDX-3.0 content characterizing details about a product. This could include details of the content and composition of the product, provenence details of the product and/or its composition, licensing information, known quality or security issues, etc.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/Bom`

Name	Bom
Instantiability	Concrete
SubclassOf	Bundle

Properties

Property	Type	minCount	maxCount

Bundle

Summary

A collection of Elements that have a shared context.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/Bundle`

Name	Bundle
Instantiability	Concrete
SubclassOf	ElementCollection

Properties

Property	Type	minCount	maxCount
context	xsd:string	0	1

CreationInfo

Summary

Provides information about the creation of the Element.

Description

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

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/CreationInfo`

Name	CreationInfo
Instantiability	Concrete

Properties

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

DictionaryEntry

Summary

A key with an associated value.

Description

The class used for implementing a generic string mapping (also known as associative array, dictionary, or hash map) in SPDX. Each DictionaryEntry contains a key-value pair which maps the key to its associated value. To implement a dictionary, this class is to be used in a collection with unique keys.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/DictionaryEntry`

Name	DictionaryEntry
Instantiability	Concrete

Properties

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

Element

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/Element`

Name	Element
Instantiability	Abstract

Properties

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

ElementCollection

Summary

A collection of Elements, not necessarily with unifying context.

Description

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

Note that all ElementCollections must conform to the core profile even if the core profile is no specified in the profileConformance property. If the profileConformance property is not provided, core is to be assumed as the default.

Constraints If the ElementCollection has at least 1 element, it must also have at least 1 rootElement.

The element must not be of type SpdxDocument.

The rootElement must not be of type SpdxDocument.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/ElementCollection`

Name	ElementCollection
Instantiability	Abstract
SubclassOf	Element

Properties

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

ExternalIdentifier

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/ExternalIdentifier`

Name	ExternalIdentifier
Instantiability	Concrete

Properties

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

ExternalMap

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/ExternalMap`

Name	ExternalMap
Instantiability	Concrete

Properties

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

ExternalRef

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/ExternalRef`

Name	ExternalRef
Instantiability	Concrete

Properties

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

Hash

Summary

A mathematically calculated representation of a grouping of data.

Description

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

This is commonly used for integrity checking of data.

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/Hash`

Name	Hash
Instantiability	Concrete
SubclassOf	IntegrityMethod

Properties

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

IntegrityMethod

Summary

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

Description

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

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/IntegrityMethod`

Name	IntegrityMethod
Instantiability	Abstract

Properties

Property	Type	minCount	maxCount
comment	xsd:string	0	1

LifecycleScopedRelationship

Summary

Provide context for a relationship that occurs in the lifecycle.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/LifecycleScopedRelationship`

Name	LifecycleScopedRelationship
Instantiability	Concrete
SubclassOf	Relationship

Properties

Property	Type	minCount	maxCount
scope	LifecycleScopeType	0	1

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/ (TODO: update the URLs as soon as the context is publicly available)

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

1) An SPDX content producer wishing to provide clarity of their serialization of an SPDX 2.X simple style collection where all content is newly minted and a single prefix-namespace is used. The consumer of SPDX content wishes to preserve the name space mapping provided by such a producer. In this case, the consumer would record the namespace map prefixes in the NamespaceMap such that subsequent serializations could reproduce the prefixes / namespaces in the native serialization format. 2) An SPDX content producer wishing to maintain consistent prefix use and understanding across multiple different serialization formats of the produced content. For example, an SBOM producer wishes to share/publish the SBOM as JSON-LD and XML. The producer can specify the preferred prefix mappings in the native serialization format using information from a single Namespacemap accessible local to the producer. 3) An SPDX content consumer/producer wishing to maintain consistent prefix use while round tripping from SPDX content received, deserialized, modified/extended in some way, and then reserialized in the same serialization form. In this case the prefix-namespace mappings utilized in the content are transformed from the original native namespace/prefix into the in memory NamespaceMap then transformed from the NamespaceMap back into the resultant serialization native namespace / prefix format.

Organization

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/Organization`

Name	Organization
Instantiability	Concrete
SubclassOf	Agent

Properties

Property	Type	minCount	maxCount

PackageVerificationCode

Summary

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

Description

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

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

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

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

Algorithm:

```
verificationcode = 0
filelist = templist = ""
for all files in the package {
     if file is an "excludes" file, skip it /* exclude SPDX analysis file(s) */
     else append templist with "algorithm(file)/n"
    }

 sort templist in ascending order by algorithm value

 filelist = templist with "/n"s removed. /* ordered sequence of algorithm values with no separators

 hashValue = algorithm(filelist) /* Where algorithm(file) applies a hash algorithm on the contents
```

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

Person

Summary

An individual human being.

Description

A Person is an individual human being.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/Person`

Name	Person
Instantiability	Concrete
SubclassOf	Agent

Properties

Property	Type	minCount	maxCount

PositiveIntegerRange

Summary

A tuple of two positive integers that define a range.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/PositiveIntegerRange`

Name	PositiveIntegerRange
Instantiability	Concrete

Properties

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

Relationship

Summary

Describes a relationship between one or more elements.

Description

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

To explicitly assert that no such relationships exist, the `to` property should contain the 'NONE' individual and no other elements. A relationship that contains 'NONE' and additional elements in the `to` property is not valid.

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/Relationship`

Name	Relationship
Instantiability	Concrete
SubclassOf	Element

Properties

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

SoftwareAgent

Summary

A software agent.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/SoftwareAgent`

Name	SoftwareAgent
Instantiability	Concrete
SubclassOf	Agent

Properties

Property	Type	minCount	maxCount

SpdxDocument

Summary

A collection of SPDX Elements that could potentially be serialized.

Description

The SpdxDocument provides a convenient way to express information about collections of SPDX Elements that could potentially be serialized as complete units (e.g., all in-scope SPDX data within a single JSON-LD file). SpdxDocument is independent of any particular serialization format or instance. Information we wish to preserve about a specific instance of serialization of this SPDX content is NOT expressed using the SpdxDocument but rather using an associated Artifact representing a particular instance of SPDX data physical serialization.

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/SpdxDocument`

Name	SpdxDocument
Instantiability	Concrete
SubclassOf	ElementCollection

Properties

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

Tool

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/Tool`

Name	Tool
Instantiability	Concrete
SubclassOf	Element

Properties

Property	Type	minCount	maxCount

algorithm

Summary

Specifies the algorithm used for calculating the hash value.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/algorithm`

Name	algorithm
Nature	ObjectProperty
Range	HashAlgorithm

Referenced

- /Core/Hash
- /Core/PackageVerificationCode

annotationType

Summary

Describes the type of annotation.

Description

An annotationType describes the type of an annotation.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/annotationType`

Name	annotationType
Nature	ObjectProperty
Range	AnnotationType

Referenced

- /Core/Annotation

beginIntegerRange

Summary

Defines the beginning of a range.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/beginIntegerRange`

Name	beginIntegerRange
Nature	DataProperty
Range	xsd:positiveInteger

Referenced

- /Core/PositiveIntegerRange

builtTime

Summary

Specifies the time an artifact was built.

Description

A builtTime specifies the time an artifact was built.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/builtTime`

Name	builtTime
Nature	DataProperty
Range	DateTime

Referenced

- /Core/Artifact

comment

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/comment`

Name	comment
Nature	DataProperty
Range	xsd:string

Referenced

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

completeness

Summary

Provides information about the completeness of relationships.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/completeness`

Name	completeness
Nature	ObjectProperty
Range	RelationshipCompleteness

Referenced

- /Core/Relationship

contentType

Summary

Specifies the media type of an Element or Property.

Description

ContentType specifies the media type of an Element or Property.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/contentType`

Name	contentType
Nature	DataProperty
Range	MediaType

Referenced

- /Core/Annotation
- /Core/ExternalRef

context

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/context`

Name	context
Nature	DataProperty
Range	xsd:string

Referenced

- /Core/Bundle

created

Summary

Identifies when the Element was originally created.

Description

Created is a date that identifies when the Element was originally created. The time stamp can serve as an indication as to whether the analysis needs to be updated. This is often the date of last change (e.g., a git commit date), not the date when the SPDX data was created, as doing so supports reproducible builds.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/created`

Name	created
Nature	DataProperty
Range	DateTime

Referenced

- /Core/CreationInfo

createdBy

Summary

Identifies who or what created the Element.

Description

CreatedBy identifies who or what created the Element. The generation method will assist the recipient of the Element in assessing the general reliability/accuracy of the analysis information.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/createdBy`

Name	createdBy
Nature	ObjectProperty
Range	Agent

Referenced

- /Core/CreationInfo

createdUsing

Summary

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

Description

CreatedUsing identifies the tooling that was used during the creation of the Element. The generation method will assist the recipient of the Element in assessing the general reliability/ accuracy of the analysis information.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/createdUsing`

Name	createdUsing
Nature	ObjectProperty
Range	Tool

Referenced

- /Core/CreationInfo

creationInfo

Summary

Provides information about the creation of the Element.

Description

CreationInfo provides information about the creation of the Element.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/creationInfo`

Name	creationInfo
Nature	ObjectProperty
Range	CreationInfo

Referenced

- /Core/Element

dataLicense

Summary

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

Description

The data license provides the license under which the SPDX documentation of the Element can be used. This is to alleviate any concern that content (the data or database) in an SPDX file is subject to any form of intellectual property right that could restrict the re-use of the information or the creation of another SPDX file for the same project(s). This approach avoids intellectual property and related restrictions over the SPDX file, however individuals can still contract with each other to restrict release of specific collections of SPDX files (which map to software bill of materials) and the identification of the supplier of SPDX files. Compliance with this document includes populating the SPDX fields therein with data related to such fields ("SPDX-Metadata"). This document contains numerous fields where an SPDX file creator may provide relevant explanatory text in SPDX-Metadata. Without opining on the lawfulness of "database rights" (in jurisdictions where applicable), such explanatory text is copyrightable subject matter in most Berne Convention countries. By using the SPDX specification, or any portion hereof, you hereby agree that any copyright rights (as determined by your jurisdiction) in any SPDX-Metadata, including without limitation explanatory text, shall be subject to the terms of the Creative Commons CC0 1.0 Universal license. For SPDX-Metadata not containing any copyright rights, you hereby agree and acknowledge that the SPDX-Metadata is provided to you "as-is" and without any representations or warranties of any kind concerning the SPDX-Metadata, express, implied, statutory or otherwise, including without limitation warranties of title, merchantability, fitness for a particular purpose, non-infringement, or the absence of latent or other defects, accuracy, or the presence or absence of errors, whether or not discoverable, all to the greatest extent permissible under applicable law.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/dataLicense`

Name	dataLicense

definingArtifact

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/definingArtifact`

Name	definingArtifact
Nature	ObjectProperty
Range	Artifact

Referenced

- /Core/ExternalMap

description

Summary

Provides a detailed description of the Element.

Description

This field is a detailed description of the Element. It may also be extracted from the Element itself. The intent is to provide recipients of the SPDX file with a detailed technical explanation of the functionality, anticipated use, and anticipated implementation of the Element. This field may also include a description of improvements over prior versions of the Element.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/description`

Name	description
Nature	DataProperty
Range	xsd:string

Referenced

- /Core/Element

element

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/element`

Name	element
Nature	ObjectProperty
Range	Element

Referenced

- /Core/ElementCollection

endIntegerRange

Summary

Defines the end of a range.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/endIntegerRange`

Name	endIntegerRange
Nature	DataProperty
Range	xsd:positiveInteger

Referenced

- /Core/PositiveIntegerRange

endTime

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/endTime`

Name	endTime
Nature	DataProperty
Range	DateTime

Referenced

- /Core/Relationship

extension

Summary

Specifies an Extension characterization of some aspect of an Element.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/extension`

Name	extension
Nature	ObjectProperty
Range	/Extension/Extension

Referenced

- /Core/Element

externalIdentifier

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/externalIdentifier`

Name	externalIdentifier
Nature	ObjectProperty
Range	ExternalIdentifier

Referenced

- /Core/Element

externalIdentifierType

Summary

Specifies the type of the external identifier.

Description

An externalIdentifierType specifies the type of the external identifier.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/externalIdentifierType`

Name	externalIdentifierType
Nature	ObjectProperty
Range	ExternalIdentifierType

Referenced

- /Core/ExternalIdentifier

externalRef

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/externalRef`

Name	externalRef
Nature	ObjectProperty
Range	ExternalRef

Referenced

- /Core/Element

externalRefType

Summary

Specifies the type of the external reference.

Description

An externalRefType specifies the type of the external reference.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/externalRefType`

Name	externalRefType
Nature	ObjectProperty
Range	ExternalRefType

Referenced

- /Core/ExternalRef

externalSpdxId

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/externalSpdxId`

Name	externalSpdxId
Nature	DataProperty
Range	xsd:anyURI

Referenced

- /Core/ExternalMap

from

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/from`

Name	from
Nature	ObjectProperty
Range	Element

Referenced

- /Core/Relationship

hashValue

Summary

The result of applying a hash algorithm to an Element.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/hashValue`

Name	hashValue
Nature	DataProperty
Range	xsd:string

Referenced

- /Core/Hash
- /Core/PackageVerificationCode

identifier

Summary

Uniquely identifies an external element.

Description

An identifier uniquely identifies an external element.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/identifier`

Name	identifier
Nature	DataProperty
Range	xsd:string

Referenced

- /Core/ExternalIdentifier

identifierLocator

Summary

Provides the location for more information regarding an external identifier.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/identifierLocator`

Name	identifierLocator
Nature	DataProperty
Range	xsd:anyURI

Referenced

- /Core/ExternalIdentifier

imports

Summary

Provides an ExternalMap of Element identifiers.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/imports`

Name	imports
Nature	ObjectProperty
Range	ExternalMap

Referenced

- /Core/SpdxDocument

issuingAuthority

Summary

An entity that is authorized to issue identification credentials.

Description

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

The entity may be a government, non-profit, educational institution, or commercial enterprise. The string provides a unique identifier for the issuing authority.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/issuingAuthority`

Name	issuingAuthority
Nature	DataProperty
Range	xsd:string

Referenced

- /Core/ExternalIdentifier

key

Summary

A key used in a generic key-value pair.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/key`

Name	key
Nature	DataProperty
Range	xsd:string

Referenced

- /Core/DictionaryEntry

locationHint

Summary

Provides an indication of where to retrieve an external Element.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/locationHint`

Name	locationHint
Nature	DataProperty
Range	xsd:anyURI

Referenced

- /Core/ExternalMap

locator

Summary

Provides the location of an external reference.

Description

A locator provides the location of an external reference.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/locator`

Name	locator
Nature	DataProperty
Range	xsd:string

Referenced

- /Core/ExternalRef

name

Summary

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

Description

This field identifies the name of an Element as designated by the creator. The name of an Element is an important convention and easier to refer to than the URI.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/name`

Name	name
Nature	DataProperty
Range	xsd:string

Referenced

- /Core/Element

namespace

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/namespace`

Name	namespace
Nature	DataProperty
Range	xsd:anyURI

Referenced

- /Core/NamespaceMap

namespaceMap

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/namespaceMap`

Name	namespaceMap
Nature	ObjectProperty
Range	NamespaceMap

Referenced

- /Core/SpdxDocument

originatedBy

Summary

Identifies from where or whom the Element originally came.

Description

OriginatedBy identifies from where or whom the Element originally came.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/originatedBy`

Name	originatedBy
Nature	ObjectProperty
Range	Agent

Referenced

- /Core/Artifact

packageVerificationCodeExcludedFile

Summary

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

Description

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

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/packageVerificationCodeExcludedFile`

Name	packageVerificationCodeExcludedFile
Nature	DataProperty
Range	xsd:string

Referenced

- /Core/PackageVerificationCode

prefix

Summary

A substitute for a URI.

Description

A prefix is a substitute for a URI.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/prefix`

Name	prefix
Nature	DataProperty
Range	xsd:string

Referenced

- /Core/NamespaceMap

profileConformance

Summary

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

Description

Describes a profile to which the creator of this ElementCollection intends to conform. The profileConformance will apply to all Elements contained within the collection as well as the collection itself. Conformance to a profile is defined by the additional restrictions documented in the profile specific documentation and schema files. Use of this property allows the creator of an ElementCollection to communicate to consumers their intent to adhere to the profile additional restrictions. The profileConformance has a default value of core if no other profileConformance is specified since all ElementCollections and Element must adhere to the core profile.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/profileConformance`

Name	profileConformance
Nature	ObjectProperty
Range	ProfileIdentifierType

Referenced

- /Core/ElementCollection

relationshipType

Summary

Information about the relationship between two Elements.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/relationshipType`

Name	relationshipType
Nature	ObjectProperty
Range	RelationshipType

Referenced

- /Core/Relationship

releaseTime

Summary

Specifies the time an artifact was released.

Description

A releaseTime specifies the time an artifact was released.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/releaseTime`

Name	releaseTime
Nature	DataProperty
Range	DateTime

Referenced

- /Core/Artifact

rootElement

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/rootElement`

Name	rootElement
Nature	ObjectProperty
Range	Element

Referenced

- /Core/ElementCollection

scope

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/scope`

Name	scope
Nature	ObjectProperty
Range	LifecycleScopeType

Referenced

- /Core/LifecycleScopedRelationship

spdxId

Summary

Identifies an Element to be referenced by other Elements.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/spdxId`

Name	spdxId
Nature	DataProperty
Range	xsd:anyURI

Referenced

- /Core/Element

specVersion

Summary

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

Description

The specVersion provides a reference number that can be used to understand how to parse and interpret an Element. It will enable both future changes to the specification and to support backward compatibility. The major version number shall be incremented when incompatible changes between versions are made (one or more sections are created, modified or deleted). The minor version number shall be incremented when backwards compatible changes are made.

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/specVersion`

Name	specVersion
Nature	DataProperty
Range	SemVer

Referenced

- /Core/CreationInfo

standardName

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/standardName`

Name	standardName
Nature	DataProperty
Range	xsd:string

Referenced

- /Core/Artifact

startTime

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/startTime`

Name	startTime
Nature	DataProperty
Range	DateTime

Referenced

- /Core/Relationship

statement

Summary

Commentary on an assertion that an annotator has made.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/statement`

Name	statement
Nature	DataProperty
Range	xsd:string

Referenced

- /Core/Annotation

subject

Summary

An Element an annotator has made an assertion about.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/subject`

Name	subject
Nature	ObjectProperty
Range	Element

Referenced

- /Core/Annotation

summary

Summary

A short description of an Element.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/summary`

Name	summary
Nature	DataProperty
Range	xsd:string

Referenced

- /Core/Element

suppliedBy

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/suppliedBy`

Name	suppliedBy
Nature	ObjectProperty
Range	Agent

Referenced

- /Core/Artifact
- /Security/VulnAssessmentRelationship

supportLevel

Summary

Specifies the level of support associated with an artifact.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/supportLevel`

Name	supportLevel
Nature	ObjectProperty
Range	SupportType

Referenced

- /Core/Artifact

to

Summary

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

Description

This field references an Element on the right-hand side of a relationship. If it is not provided, it indicates that there are no known relationships of the given type.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/to`

Name	to
Nature	ObjectProperty
Range	Element

Referenced

- /Core/Relationship

validUntilTime

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/validUntilTime`

Name	validUntilTime
Nature	DataProperty
Range	DateTime

Referenced

- /Core/Artifact

value

Summary

A value used in a generic key-value pair.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/value`

Name	value
Nature	DataProperty
Range	xsd:string

Referenced

- /Core/DictionaryEntry

verifiedUsing

Summary

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

Description

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

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/verifiedUsing`

Name	verifiedUsing
Nature	ObjectProperty
Range	IntegrityMethod

Referenced

- /Core/Element
- /Core/ExternalMap

AnnotationType

Summary

Specifies the type of an annotation.

Description

AnnotationType specifies the type of an annotation.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/AnnotationType`

Name	AnnotationType

Entries

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

ExternalIdentifierType

Summary

Specifies the type of an external identifier.

Description

ExteralIdentifierType specifies the type of an external identifier.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/ExternalIdentifierType`

Name	ExternalIdentifierType

Entries

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

ExternalRefType

Summary

Specifies the type of an external reference.

Description

ExternalRefType specifies the type of an external reference.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/ExternalRefType`

Name	ExternalRefType

Entries

- altDownloadLocation: A reference to an alternative download location.
- altWebPage: A reference to an alternative web page.
- binaryArtifact: A reference to binary artifacts related to a package.
- bower: A reference to a bower package.
- buildMeta: A reference build metadata related to a published package.
- buildSystem: A reference build system used to create or publish the package.
- certificationReport: A reference to a certification report for a package from an accredited/independent body.
- chat: A reference to the instant messaging system used by the maintainer for a package.
- componentAnalysisReport: A reference to a Software Composition Analysis (SCA) report.
- cwe: A reference to a source of software flaw defined within the official CWE Dictionary that conforms to the CWE specification as defined by https://csrc.nist.gov/glossary/term/common_weakness_enumeration.
- 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.

HashAlgorithm

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/HashAlgorithm`

Name	HashAlgorithm

Entries

- blake2b256: blake2b algorithm with a digest size of 256 https://datatracker.ietf.org/doc/html/rfc7693#section-4
- blake2b384: blake2b algorithm with a digest size of 384 https://datatracker.ietf.org/doc/html/rfc7693#section-4
- blake2b512: blake2b algorithm with a digest size of 512 https://datatracker.ietf.org/doc/html/rfc7693#section-4
- blake3: https://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

LifecycleScopeType

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/LifecycleScopeType`

Name	LifecycleScopeType

Entries

- build: A relationship has specific context implications during an element's build phase, during development.
- design: A relationship has specific context implications during an element's design.
- development: A relationship has specific context implications during development phase of an element.
- other: A relationship has other specific context information necessary to capture that the above set of enumerations does not handle.
- runtime: A relationship has specific context implications during the execution phase of an element.
- test: A relationship has specific context implications during an element's testing phase, during development.

PresenceType

Summary

Categories of presence or absence.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/PresenceType`

Name	PresenceType

Entries

- no: Indicates absence of the field.
- noAssertion: Makes no assertion about the field.
- yes: Indicates presence of the field.

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/3.0.0/terms/Core/ProfileIdentifierType`

Name	ProfileIdentifierType

Entries

- ai: the element follows the AI profile specification
- build: the element follows the Build profile specification
- core: the element follows the Core profile specification
- dataset: the element follows the Dataset profile specification
- expandedLicensing: the element follows the 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

RelationshipCompleteness

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/RelationshipCompleteness`

Name	RelationshipCompleteness

Entries

- complete: The relationship is known to be exhaustive.
- incomplete: The relationship is known not to be exhaustive.
- noAssertion: No assertion can be made about the completeness of the relationship.

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/3.0.0/terms/Core/RelationshipType`

Name	RelationshipType

Entries

- affects: (Security/VEX) The `from` vulnerability affect each `to` Element
- amendedBy: The `from` Element is amended by each `to` Element
- ancestorOf: The `from` Element is an ancestor of each `to` Element
- availableFrom: The `from` Element is available from the additional supplier described by each `to` Element
- configures: The `from` Element is a configuration applied to each `to` Element during a LifecycleScopeType period
- contains: The `from` Element contains each `to` Element
- coordinatedBy: (Security) The `from` Vulnerability is coordinatedBy the `to` Agent(s) (vendor, researcher, or consumer agent)

SupportType

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/SupportType`

Name	SupportType

Entries

- deployed: in addition to being supported by the supplier, the software is known to have been deployed and is in use. For a software as a service provider, this implies the software is now available as a service.
- development: the artifact is in active development and is not considered ready for formal support from the supplier.
- endOfSupport: there is a defined end of support for the artifact from the supplier. This may also be referred to as end of life. There is a validUntilDate that can be used to signal when support ends for the artifact.
- limitedSupport: the artifact has been released, and there is limited support available from the supplier. There is a validUntilDate that can provide additional information about the duration of support.
- noAssertion: no assertion about the type of support is made. This is considered the default if no other support type is used.
- noSupport: there is no support for the artifact from the supplier, consumer assumes any support obligations.
- support: the artifact has been released, and is supported from the supplier. There is a validUntilDate that can provide additional information about the duration of support.

DateTime

Summary

A string representing a specific date and time.

Description

A Datetime is a string representation of a specific date and time. It has resolution of seconds and is always expressed in UTC timezone. The specific format is one of the most commonly used ISO-8601 formats.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/DateTime`

Name	DateTime
SubclassOf	xsd:dateTimeStamp

Format

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

MediaType

Summary

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

Description

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

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/MediaType`

Name	MediaType
SubclassOf	xsd:string

Format

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

SemVer

Summary

A string constrained to the SemVer 2.0.0 specification.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/SemVer`

Name	SemVer
SubclassOf	xsd:string

Format

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

NoAssertionElement

Summary

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

Description

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

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/NoAssertionElement`

Name	NoAssertionElement
Type	Element
IRI	``

NoneElement

Summary

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

Description

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

For example, a Relationship with `from` =Element1, `relationshipType` ="ancestorOf", and `to` =NONE is explicitly expressing an assertion that Element1 has no descendents.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Core/NoneElement`

Name	NoneElement
Type	Element
IRI	``

Software

Summary

Everything having to do with software.

Description

The Software namespace defines concepts related to software artifacts.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software`

Name	Software

ContentIdentifier

Summary

A canonical, unique, immutable identifier

Description

A ContentIdentifier is a canonical, unique, immutable identifier of the content of a software artifact, such as a package, a file, or a snippet. It can be used for verifying its identity and integrity.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/ContentIdentifier`

Name	ContentIdentifier
Instantiability	Concrete
SubclassOf	/Core/IntegrityMethod

Properties

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

File

Summary

Refers to any object that stores content on a computer.

Description

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

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/File`

Name	File
Instantiability	Concrete
SubclassOf	/Software/SoftwareArtifact

Properties

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

Package

Summary

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

Description

A package refers to any unit of content that can be associated with a distribution of software.
Typically, a package is composed of one or more files.
Any of the following non-limiting examples may be (but are not required to be) represented in SPDX as a package:

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

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/Package`

Name	Package
Instantiability	Concrete
SubclassOf	/Software/SoftwareArtifact

Properties

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

Sbom

Summary

A collection of SPDX Elements describing a single package.

Description

A Software Bill of Materials (SBOM) is a collection of SPDX Elements describing a single package. This could include details of the content and composition of the product, provenance details of the product and/or its composition, licensing information, known quality or security issues, etc.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/Sbom`

Name	Sbom
Instantiability	Concrete
SubclassOf	/Core/Bom

Properties

Property	Type	minCount	maxCount
sbomType	SbomType	0	*

Snippet

Summary

Describes a certain part of a file.

Description

A Snippet describes a certain part of a file and can be used when the file is known to have some content that has been included from another original source. Snippets are useful for denoting when part of a file may have been originally created under another license or copied from a place with a known vulnerability.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/Snippet`

Name	Snippet
Instantiability	Concrete
SubclassOf	/Software/SoftwareArtifact

Properties

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

SoftwareArtifact

Summary

A distinct article or unit related to Software.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/SoftwareArtifact`

Name	SoftwareArtifact
Instantiability	Abstract
SubclassOf	/Core/Artifact

Properties

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

additionalPurpose

Summary

Provides additional purpose information of the software artifact.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/additionalPurpose`

Name	additionalPurpose
Nature	ObjectProperty
Range	SoftwarePurpose

Referenced

- /Software/SoftwareArtifact

attributionText

Summary

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

Description

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

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

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

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/attributionText`

Name	attributionText
Nature	DataProperty
Range	xsd:string

Referenced

- /Software/SoftwareArtifact

byteRange

Summary

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

Description

This field defines the byte range in the original host file that the snippet information applies to. A range of bytes is independent of various formatting concerns, and the most accurate way of referring to the differences. The choice was made to start the numbering of the byte range at 1 to be consistent with the W3C pointer method vocabulary.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/byteRange`

Name	byteRange
Nature	DataProperty
Range	/Core/PositiveIntegerRange

Referenced

- /Software/Snippet

contentIdentifier

Summary

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

Description

A contentIdentifier is a canonical, unique, immutable identifier of the content of a software artifact, such as a package, a file, or a snippet. It may be used for verifying its identity and/or integrity.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/contentIdentifier`

Name	contentIdentifier
Nature	DataProperty
Range	ContentIdentifier

Referenced

- /Software/SoftwareArtifact

contentIdentifierType

Summary

Specifies the type of the content identifier.

Description

A contentIdentifierType specifies the type of the content identifier.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/contentIdentifierType`

Name	contentIdentifierType
Nature	ObjectProperty
Range	ContentIdentifierType

Referenced

- /Software/ContentIdentifier

contentIdentifierValue

Summary

Specifies the value of the content identifier.

Description

A contentIdentifierValue specifies the value of a content identifier.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/contentIdentifierValue`

Name	contentIdentifierValue
Nature	DataProperty
Range	xsd:anyURI

Referenced

- /Software/ContentIdentifier

contentType

Summary

Provides information about the content type of an Element.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/contentType`

Name	contentType
Nature	DataProperty
Range	/Core/MediaType

Referenced

- /Software/File

copyrightText

Summary

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

Description

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

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

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

If a copyrightText has a "NOASSERTION" value, this indicates that one of the following applies: * the SPDX data creator has attempted to but cannot reach a reasonable objective determination; * the SPDX data creator has made no attempt to determine this field; or * the SPDX data creator has intentionally provided no information (no meaning should be implied by doing so).

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/copyrightText`

Name	copyrightText
Nature	DataProperty
Range	xsd:string

downloadLocation

Summary

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

Description

DownloadLocation identifies the download Uniform Resource Identifier for the package at the time that the document was created. Where and how to download the exact package being referenced is critical for verification and tracking data.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/downloadLocation`

Name	downloadLocation
Nature	DataProperty
Range	xsd:anyURI

Referenced

- /Software/Package

fileKind

Summary

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

Description

An SPDX file may represent a specific file or a directory of files. In the future, this may be extended to other kinds (e.g. network based files).

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/fileKind`

Name	fileKind
Nature	ObjectProperty
Range	FileKindType

Referenced

- /Software/File

homePage

Summary

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

Description

HomePage is a place for the SPDX document creator to record a website that serves as the package's home page. This saves the recipient of the SPDX document who is looking for more info from having to search for and verify a match between the package and the associated project home page. This link can also be used to reference further information about the package referenced by the SPDX document creator.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/homePage`

Name	homePage
Nature	DataProperty
Range	xsd:anyURI

Referenced

- /Software/Package

lineRange

Summary

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

Description

This field defines the line range in the original host file that the snippet information applies to. If there is a disagreement between the byte range and line range, the byte range values will take precedence. A range of lines is a convenient reference for those files where there is a known line delimiter. The choice was made to start the numbering of the lines at 1 to be consistent with the W3C pointer method vocabulary.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/lineRange`

Name	lineRange
Nature	DataProperty
Range	/Core/PositiveIntegerRange

Referenced

- /Software/Snippet

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

the purl URL string is defined by seven components:

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

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

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/packageUrl`

Name	packageUrl
Nature	DataProperty
Range	xsd:anyURI

packageVersion

Summary

Identify the version of a package.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/packageVersion`

Name	packageVersion
Nature	DataProperty
Range	xsd:string

Referenced

- /Software/Package

primaryPurpose

Summary

Provides information about the primary purpose of the software artifact.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/primaryPurpose`

Name	primaryPurpose
Nature	ObjectProperty
Range	SoftwarePurpose

Referenced

- /Software/SoftwareArtifact

sbomType

Summary

Provides information about the type of an SBOM.

Description

This field is a reasonable estimation of the type of SBOM created from a creator perspective. It is intended to be used to give guidance on the elements that may be contained within it. Aligning with the guidance produced in Types of Software Bill of Material (SBOM) Documents.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/sbomType`

Name	sbomType
Nature	ObjectProperty
Range	SbomType

Referenced

- /Software/Sbom

snippetFromFile

Summary

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

Description

The field identifies the file which contains the snippet.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/snippetFromFile`

Name	snippetFromFile
Nature	ObjectProperty
Range	File

Referenced

- /Software/Snippet

sourceInfo

Summary

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

Description

SourceInfo records any relevant background information or additional comments about the origin of the package. For example, this field might include comments indicating whether the package was pulled from a source code management system or has been repackaged. The creator can provide additional information to describe any anomalies or discoveries in the determination of the origin of the package.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/sourceInfo`

Name	sourceInfo
Nature	DataProperty
Range	xsd:string

Referenced

- /Software/Package

ContentIdentifierType

Summary

Specifies the type of a content identifier.

Description

ContentIdentifierType specifies the type of a content identifier.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/ContentIdentifierType`

Name	ContentIdentifierType

Entries

- gitoid: Gitoid stands for Git Object ID and a gitoid of type blob is a unique hash of a binary artifact. A gitoid may represent the software Artifact ID or the OmniBOR Identifier for the software artifact's associated OmniBOR Document.
- swhid: SoftWare Hash IDentifier, persistent intrinsic identifiers for digital artifacts. The syntax of the identifiers is defined in the SWHID specification and in the case of filess they typically look like `swh:1:cnt:94a9ed024d3859793618152ea559a168bbcbb5e2`.

FileKindType

Summary

Enumeration of the different kinds of SPDX file.

Description

An SPDX file may represent a file on disk or a directory of files. In the future, this may be extended to other kinds (e.g. network based files).

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/FileKindType`

Name	FileKindType

Entries

- directory: The file represents a directory and all content stored in that directory.
- file: The file represents a single file (default).

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/3.0.0/terms/Software/SbomType`

Name	SbomType

Entries

- analyzed: SBOM generated through analysis of artifacts (e.g., executables, packages, containers, and virtual machine images) after its build. Such analysis generally requires a variety of heuristics. In some contexts, this may also be referred to as a “3rd party” SBOM.
- build: SBOM generated as part of the process of building the software to create a releasable artifact (e.g., executable or package) from data such as source files, dependencies, built components, build process ephemeral data, and other SBOMs.
- deployed: SBOM provides an inventory of software that is present on a system. This may be an assembly of other SBOMs that combines analysis of configuration options, and examination of execution behavior in a (potentially simulated) deployment environment.
- design: SBOM of intended, planned software project or product with included components (some of which may not yet exist) for a new software artifact.
- runtime: SBOM generated through instrumenting the system running the software, to capture only components present in the system, as well as external call-outs or

SoftwarePurpose

Summary

Provides information about the primary purpose of an Element.

Description

This field provides information about the primary purpose of an Element. Software Purpose is intrinsic to how the Element is being used rather than the content of the Element. This field is a reasonable estimate of the most likely usage of the Element from the producer and consumer perspective from which both parties can draw conclusions about the context in which the Element exists.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Software/SoftwarePurpose`

Name	SoftwarePurpose

Entries

- application: the Element is a software application
- archive: the Element is an archived collection of one or more files (.tar, .zip, etc)
- bom: Element is a bill of materials
- configuration: Element is configuration data
- container: the Element is a container image which can be used by a container runtime application
- data: Element is data
- device: the Element refers to a chipset, processor, or electronic board
- deviceDriver: Element represents software that controls hardware devices
- diskImage: the Element refers to a disk image that can be written to a disk, booted in a VM, etc. A disk image typically contains most or all of the components necessary to boot, such as bootloaders, kernels, firmware, userspace, etc.
- documentation: Element is documentation
- evidence: the Element is the evidence that a specification or requirement has been fulfilled

Security

Summary

The Security Profile captures security related information.

Description

The Security Profile captures security related information.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security`

Name	Security

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

CvssV3VulnAssessmentRelationship

Summary

Provides a CVSS version 3 assessment for a vulnerability.

Description

A CvssV3VulnAssessmentRelationship relationship describes the determined score, severity, and vector of a vulnerability using version 3.0 or 3.1 of the Common Vulnerability Scoring System (CVSS). It is intended to communicate the results of using a CVSS calculator.

Constraints

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

Syntax

CvssV4VulnAssessmentRelationship

Summary

Provides a CVSS version 4 assessment for a vulnerability.

Description

A CvssV4VulnAssessmentRelationship relationship describes the determined score, severity, and vector of a vulnerability using version 4 of the Common Vulnerability Scoring System (CVSS) as defined on https://www.first.org/cvss/v4.0/specification-document. It is intented to communicate the results of using a CVSS calculator.

Constraints

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

Syntax

EpssVulnAssessmentRelationship

Summary

Provides an EPSS assessment for a vulnerability.

Description

An EpssVulnAssessmentRelationship relationship describes the likelihood or probability that a vulnerability will be exploited in the wild, and the percentile ranking of probability relative to all other vulnerabilities' EPSS scores, using the Exploit Prediction Scoring System (EPSS) as defined at https://www.first.org/epss/model.

Constraints

- The relationship type must be set to hasAssessmentFor.
- The probability must be between 0 and 1.
- The percentile must be between 0 and 1.

Syntax

```
{
  "@type": "EpssVulnAssessmentRelationship",
  "@id": "urn:spdx.dev:epss-CVE-2020-28498",
  "relationshipType": "hasAssessmentFor",
  "probability": 0.00105,
  "percentile": 0.42356,
  "from": "urn:spdx.dev:vuln-cve-2020-28498",
  "to": ["urn:product-acme-application-1.3"],
  "suppliedBy": ["urn:spdx.dev:agent-jane-doe"],
  "publishedTime": "2023-10-05T00:00:30Z"
}
```

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/EpssVulnAssessmentRelationship`

Name	EpssVulnAssessmentRelationship
Instantiability	Concrete

ExploitCatalogVulnAssessmentRelationship

Summary

Provides an exploit assessment of a vulnerability.

Description

An ExploitCatalogVulnAssessmentRelationship describes if a vulnerability is listed in any exploit catalog such as the CISA Known Exploited Vulnerabilities Catalog (KEV) https://www.cisa.gov/known-exploited-vulnerabilities-catalog.

Constraints

- The relationship type must be set to hasAssessmentFor.

Syntax

```
{
  "@type": "ExploitCatalogVulnAssessmentRelationship",
  "@id": "urn:spdx.dev:exploit-catalog-1",
  "relationshipType": "hasAssessmentFor",
  "catalogType": "kev",
  "locator": "https://www.cisa.gov/known-exploited-vulnerabilities-catalog",
  "exploited": "true",
  "from": "urn:spdx.dev:vuln-cve-2023-2136",
  "to": ["urn:product-google-chrome-112.0.5615.136"],
  "suppliedBy": ["urn:spdx.dev:agent-jane-doe"],
  "publishedTime": "2021-03-09T11:04:53Z"
}
```

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/ExploitCatalogVulnAssessmentRelationship`

Name	ExploitCatalogVulnAssessmentRelationship
Instantiability	Concrete
SubclassOf	VulnAssessmentRelationship

SsvcVulnAssessmentRelationship

Summary

Provides an SSVC assessment for a vulnerability.

Description

An SsvcVulnAssessmentRelationship describes the decision made using the Stakeholder-Specific Vulnerability Categorization (SSVC) decision tree as defined on https://www.cisa.gov/stakeholder-specific-vulnerability-categorization-ssvc. It is intended to communicate the results of using the CISA SSVC Calculator.

Constraints

- The relationship type must be set to hasAssessmentFor.

Syntax

```
{
  "@type": "SsvcVulnAssessmentRelationship",
  "@id": "urn:spdx.dev:ssvc-1",
  "relationshipType": "hasAssessmentFor",
  "decisionType": "act",
  "from": "urn:spdx.dev:vuln-cve-2020-28498",
  "to": ["urn:product-acme-application-1.3"],
  "assessedElement": "urn:npm-elliptic-6.5.2",
  "suppliedBy": ["urn:spdx.dev:agent-jane-doe"],
  "publishedTime": "2021-03-09T11:04:53Z"
}
```

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/SsvcVulnAssessmentRelationship`

Name	SsvcVulnAssessmentRelationship
Instantiability	Concrete
SubclassOf	VulnAssessmentRelationship

VexAffectedVulnAssessmentRelationship

Summary

Connects a vulnerability and an element designating the element as a product affected by the vulnerability.

Description

VexAffectedVulnAssessmentRelationship connects a vulnerability and a number of elements. The relationship marks these elements as products affected by the vulnerability. This relationship corresponds to the VEX affected status.

Constraints

When linking elements using a VexAffectedVulnAssessmentRelationship, the following requirements must be observed:

- Elements linked with a VulnVexAffectedAssessmentRelationship are constrained to the affects relationship type.

Syntax

```
{
  "@type": "VexAffectedVulnAssessmentRelationship",
  "@id": "urn:spdx.dev:vex-affected-1",
  "relationshipType": "affects",
  "from": "urn:spdx.dev:vuln-cve-2020-28498",
  "to": ["urn:product-acme-application-1.3"],
  "assessedElement": "urn:npm-elliptic-6.5.2",
  "actionStatement": "Upgrade to version 1.4 of ACME application.",
  "suppliedBy": ["urn:spdx.dev:agent-jane-doe"],
  "publishedTime": "2021-03-09T11:04:53Z"
}
```

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/VexAffectedVulnAssessmentRelationship`

VexFixedVulnAssessmentRelationship

Summary

Links a vulnerability and elements representing products (in the VEX sense) where a fix has been applied and are no longer affected.

Description

VexFixedVulnAssessmentRelationship links a vulnerability to a number of elements representing VEX products where a vulnerability has been fixed and are no longer affected. It represents the VEX fixed status.

Constraints

When linking elements using a VexFixedVulnAssessmentRelationship, the following requirements must be observed:

- Elements linked with a VulnVexFixedAssessmentRelationship are constrained to using the fixedIn relationship type.
- The from: end of the relationship must ve a /Security/Vulnerability classed element.

Syntax

```
{
  "@type": "VexFixedVulnAssessmentRelationship",
  "@id": "urn:spdx.dev:vex-fixed-in-1",
  "relationshipType": "fixedIn",
  "from": "urn:spdx.dev:vuln-cve-2020-28498",
  "to": ["urn:product-acme-application-1.3"],
  "assessedElement": "urn:npm-elliptic-6.5.4",
  "suppliedBy": ["urn:spdx.dev:agent-jane-doe"],
  "publishedTime": "2021-03-09T11:04:53Z"
}
```

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/VexFixedVulnAssessmentRelationship`

VexNotAffectedVulnAssessmentRelationship

Summary

Links a vulnerability and one or more elements designating the latter as products not affected by the vulnerability.

Description

VexNotAffectedVulnAssessmentRelationship connects a vulnerability and a number of elements designating them as products not affected by the vulnerability. This relationship corresponds to the VEX not_affected status.

Constraints

When linking elements using a VexNotVulnAffectedAssessmentRelationship, the following requirements must be observed:

- Relating elements with a VexNotAffectedVulnAssessmentRelationship is restricted to the doesNotAffect relationship type.
- The from: end of the relationship must be a /Security/Vulnerability classed element.
- Both impactStatement and justificationType properties have a cardinality of 0..1 making them optional. Nevertheless, to produce a valid VEX not_affected statement, one of them MUST be defined. This is specified in the Minimum Elements for VEX.

Syntax

```
{
  "@type": "VexNotAffectedVulnAssessmentRelationship",
  "@id": "urn:spdx.dev:vex-not-affected-1",
  "relationshipType": "doesNotAffect",
  "from": "urn:spdx.dev:vuln-cve-2020-28498",
  "to": ["urn:product-acme-application-1.3"],
  "assessedElement": "urn:npm-elliptic-6.5.2",
  "justificationType": "componentNotPresent",
  "impactStatement": "Not using this vulnerable part of this library.",
  "suppliedBy": ["urn:spdx.dev:agent-jane-doe"],
  "publishedTime": "2021-03-09T11:04:53Z"
}
```

VexUnderInvestigationVulnAssessmentRelationship

Summary

Designates elements as products where the impact of a vulnerability is being investigated.

Description

VexUnderInvestigationVulnAssessmentRelationship links a vulnerability to a number of products stating the vulnerability's impact on them is being investigated. It represents the VEX under_investigation status.

Constraints

When linking elements using a VexUnderInvestigationVulnAssessmentRelationship the following requirements must be observed:

- Elements linked with a VexUnderInvestigationVulnAssessmentRelationship are constrained to using the underInvestigationFor relationship type.
- The from: end of the relationship must ve a /Security/Vulnerability classed element.

Syntax

```
{
  "@type": "VexUnderInvestigationVulnAssessmentRelationship",
  "@id": "urn:spdx.dev:vex-underInvestigation-1",
  "relationshipType": "underInvestigationFor",
  "from": "urn:spdx.dev:vuln-cve-2020-28498",
  "to": ["urn:product-acme-application-1.3"],
  "assessedElement": "urn:npm-elliptic-6.5.2",
  "suppliedBy": ["urn:spdx.dev:agent-jane-doe"],
  "publishedTime": "2021-03-09T11:04:53Z"
}
```

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/VexUnderInvestigationVulnAssessmentRelationship`

Name	VexUnderInvestigationVulnAssessmentRelationship

VexVulnAssessmentRelationship

Summary

Asbtract ancestor class for all VEX relationships

Description

VexVulnAssessmentRelationship is an abstract subclass that defined the common properties shared by all the SPDX-VEX status relationships.

Constraints

When linking elements using a VexVulnAssessmentRelationship, the following requirements must be observed:

- The from: end must be a /Security/Vulnerability classed element
- The to: end must point to elements representing the VEX *products*. To specify a different element where the vulnerability was detected, the VEX relationship can optionally specify *subcomponents* using the assessedElement property.

VEX inherits information from the document level down to its statements. When a statement is missing information it can be completed by reading the equivalent field from the containing document. For example, if a VEX relationship is missing data in its createdBy property, tools must consider the entity listed in the CreationInfo section of the document as the VEX author. In the same way, when a VEX relationship does not have a created property, the document's date must be considered as authoritative.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/VexVulnAssessmentRelationship`

Name	VexVulnAssessmentRelationship
Instantiability	Abstract
SubclassOf	VulnAssessmentRelationship

VulnAssessmentRelationship

Summary

Abstract ancestor class for all vulnerability assessments

Description

VulnAssessmentRelationship is the ancestor class common to all vulnerability assessment relationships. It factors out the common properties shared by them.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/VulnAssessmentRelationship`

Name	VulnAssessmentRelationship
Instantiability	Abstract
SubclassOf	/Core/Relationship

Properties

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

Vulnerability

Summary

Specifies a vulnerability and its associated information.

Description

Specifies a vulnerability and its associated information.

Syntax

actionStatement

Summary

Provides advise on how to mitigate or remediate a vulnerability when a VEX product is affected by it.

Description

When an element is referenced with a VexAffectedVulnAssessmentRelationship, the relationship MUST include one actionStatement that SHOULD describe actions to remediate or mitigate the vulnerability.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/actionStatement`

Name	actionStatement
Nature	DataProperty
Range	xsd:string

Referenced

- /Security/VexAffectedVulnAssessmentRelationship

actionStatementTime

Summary

Records the time when a recommended action was communicated in a VEX statement to mitigate a vulnerability.

Description

When a VEX statement communicates an affected status, the author MUST include an action statement with a recommended action to help mitigate the vulnerability's impact. The actionStatementTime property records the time when the action statement was first communicated.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/actionStatementTime`

Name	actionStatementTime
Nature	DataProperty
Range	/Core/DateTime

Referenced

- /Security/VexAffectedVulnAssessmentRelationship

assessedElement

Summary

Specifies an element contained in a piece of software where a vulnerability was found.

Description

Specifies subpackages, files or snippets referenced by a security assessment to specify the precise location where a vulnerability was found.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/assessedElement`

Name	assessedElement
Nature	ObjectProperty
Range	/Core/Element

Referenced

- /Security/VulnAssessmentRelationship

catalogType

Summary

Specifies the exploit catalog type.

Description

A catalogType is a mandatory value and must select one of the existing entries in the `ExploitCatalogType.md` vocabulary.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/catalogType`

Name	catalogType
Nature	ObjectProperty
Range	ExploitCatalogType

Referenced

- /Security/ExploitCatalogVulnAssessmentRelationship

decisionType

Summary

Provide the enumeration of possible decisions in the Stakeholder-Specific Vulnerability Categorization (SSVC) decision tree https://www.cisa.gov/sites/default/files/publications/cisa-ssvc-guide%20508c.pdf

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/decisionType`

Name	decisionType
Nature	ObjectProperty
Range	SsvcDecisionType

Referenced

- /Security/SsvcVulnAssessmentRelationship

exploited

Summary

Describe that a CVE is known to have an exploit because it's been listed in an exploit catalog.

Description

This field is set when a CVE is listed in an exploit catalog.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/exploited`

Name	exploited
Nature	DataProperty
Range	xsd:boolean

Referenced

- /Security/ExploitCatalogVulnAssessmentRelationship

impactStatement

Summary

Explains why a VEX product is not affected by a vulnerability. It is an alternative in VexNotAffectedVulnAssessmentRelationship to the machine-readable justification label.

Description

When a VEX product element is related with a VexNotAffectedVulnAssessmentRelationship and a machine readable justification label is not provided, then an impactStatement that further explains how or why the prouct(s) are not affected by the vulnerability must be provided.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/impactStatement`

Name	impactStatement
Nature	DataProperty
Range	xsd:string

Referenced

- /Security/VexNotAffectedVulnAssessmentRelationship

impactStatementTime

Summary

Timestamp of impact statement.

Description

Specifies the time when the impact statement was recorded.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/impactStatementTime`

Name	impactStatementTime
Nature	DataProperty
Range	/Core/DateTime

Referenced

- /Security/VexNotAffectedVulnAssessmentRelationship

justificationType

Summary

Impact justification label to be used when linking a vulnerability to an element representing a VEX product with a VexNotAffectedVulnAssessmentRelationship relationship.

Description

When stating that an element is not affected by a vulnerability, the VexNotAffectedVulnAssessmentRelationship must include a justification from the machine-readable labels catalog informing the reason the element is not impacted.

impactStatement which is a string with English prose can be used instead or as complementary to the justification label, but one of both MUST be defined.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/justificationType`

Name	justificationType
Nature	ObjectProperty
Range	VexJustificationType

Referenced

- /Security/VexNotAffectedVulnAssessmentRelationship

locator

Summary

Provides the location of an exploit catalog.

Description

A locator provides the location of an exploit catalog.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/locator`

Name	locator
Nature	DataProperty
Range	xsd:anyURI

Referenced

- /Security/ExploitCatalogVulnAssessmentRelationship

modifiedTime

Summary

Specifies a time when a vulnerability assessment was modified

Description

Specifies a time when a vulnerability assessment was last modified.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/modifiedTime`

Name	modifiedTime
Nature	DataProperty
Range	/Core/DateTime

Referenced

- /Security/VulnAssessmentRelationship
- /Security/Vulnerability

percentile

Summary

The percentile of the current probability score.

Description

The percentile between 0 and 1 (0 and 100%) of the current probability score, the proportion of all scored vulnerabilities with the same or a lower EPSS score. https://www.first.org/epss/data_stats

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/percentile`

Name	percentile
Nature	DataProperty
Range	xsd:decimal

Referenced

- /Security/EpssVulnAssessmentRelationship

probability

Summary

A probability score between 0 and 1 of a vulnerability being exploited.

Description

The probability score between 0 and 1 (0 and 100%) estimating the likelihood of exploitation in the wild in the next 30 days (following score publication). https://www.first.org/epss/data_stats

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/probability`

Name	probability
Nature	DataProperty
Range	xsd:decimal

Referenced

- /Security/EpssVulnAssessmentRelationship

publishedTime

Summary

Specifies the time when a vulnerability was published.

Description

Specifies the time when a vulnerability was first published.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/publishedTime`

Name	publishedTime
Nature	DataProperty
Range	/Core/DateTime

Referenced

- /Security/EpssVulnAssessmentRelationship
- /Security/VulnAssessmentRelationship
- /Security/Vulnerability

score

Summary

Provides a numerical (0-10) representation of the severity of a vulnerability.

Description

The score provides information on the severity of a vulnerability per the Common Vulnerability Scoring System as defined on https://www.first.org/cvss.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/score`

Name	score
Nature	DataProperty
Range	xsd:decimal

Referenced

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

severity

Summary

Specifies the CVSS qualitative severity rating of a vulnerability in relation to a piece of software.

Description

The severity field provides a human readable string of the resulting numerical CVSS score.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/severity`

Name	severity
Nature	ObjectProperty
Range	CvssSeverityType

Referenced

- /Security/CvssV3VulnAssessmentRelationship
- /Security/CvssV4VulnAssessmentRelationship

statusNotes

Summary

Conveys information about how VEX status was determined.

Description

A VEX statement may convey information about how status was determined and may reference other VEX information.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/statusNotes`

Name	statusNotes
Nature	DataProperty
Range	xsd:string

Referenced

- /Security/VexVulnAssessmentRelationship

vectorString

Summary

Specifies the CVSS vector string for a vulnerability.

Description

Specifies any combination of the CVSS Base, Temporal, Threat, Environmental, and/or Supplemental vector string values for a vulnerability. Supports vectorStrings specified in all CVSS versions.

Constraints

String values for the vectorString range must only include the abbreviated form of metric names specified in CVSS specifications, e.g. https://www.first.org/cvss/v4.0/specification-document#Vector-String

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/vectorString`

Name	vectorString
Nature	DataProperty
Range	xsd:string

Referenced

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

vexVersion

Summary

Specifies the version of a VEX statement.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/vexVersion`

Name	vexVersion
Nature	DataProperty
Range	xsd:string

Referenced

- /Security/VexVulnAssessmentRelationship

withdrawnTime

Summary

Specified the time and date when a vulnerability was withdrawn.

Description

Specified the time and date when a vulnerability was withdrawn.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/withdrawnTime`

Name	withdrawnTime
Nature	DataProperty
Range	/Core/DateTime

Referenced

- /Security/VulnAssessmentRelationship
- /Security/Vulnerability

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/3.0.0/terms/Security/CvssSeverityType`

Name	CvssSeverityType

Entries

- critical: When a CVSS score is between 9.0 - 10.0
- high: When a CVSS score is between 7.0 - 8.9
- low: When a CVSS score is between 0 - 3.9
- medium: When a CVSS score is between 4 - 6.9
- none: When a CVSS score is 0

ExploitCatalogType

Summary

Specifies the exploit catalog type.

Description

ExploitCatalogType specifies the type of exploit catalog that a vulnerability is listed in.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/ExploitCatalogType`

Name	ExploitCatalogType

Entries

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

SsvcDecisionType

Summary

Specifies the SSVC decision type.

Description

SsvcDecisionType specifies the type of decision that's been made according to the Stakeholder-Specific Vulnerability Categorization (SSVC) system https://www.cisa.gov/stakeholder-specific-vulnerability-categorization-ssvc

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/SsvcDecisionType`

Name	SsvcDecisionType

Entries

- act: The vulnerability requires attention from the organization's internal, supervisory-level and leadership-level individuals. Necessary actions include requesting assistance or information about the vulnerability, as well as publishing a notification either internally and/or externally. Typically, internal groups would meet to determine the overall response and then execute agreed upon actions. CISA recommends remediating Act vulnerabilities as soon as possible.
- attend: The vulnerability requires attention from the organization's internal, supervisory-level individuals. Necessary actions include requesting assistance or information about the vulnerability, and may involve publishing a notification either internally and/or externally. CISA recommends remediating Attend vulnerabilities sooner than standard update timelines.
- track: The vulnerability does not require action at this time. The organization would continue to track the vulnerability and reassess it if new information becomes available. CISA recommends remediating Track vulnerabilities within standard update timelines.
- trackStar: (Track *in the SSVC spec) The vulnerability contains specific characteristics that may require closer monitoring for changes. CISA recommends remediating Track* vulnerabilities within standard update timelines.

VexJustificationType

Summary

Specifies the VEX justification type.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Security/VexJustificationType`

Name	VexJustificationType

Entries

- componentNotPresent: The software is not affected because the vulnerable component is not in the product.
- inlineMitigationsAlreadyExist: Built-in inline controls or mitigations prevent an adversary from leveraging the vulnerability.
- vulnerableCodeCannotBeControlledByAdversary: The vulnerable component is present, and the component contains the vulnerable code. However, vulnerable code is used in such a way that an attacker cannot mount any anticipated attack.
- vulnerableCodeNotInExecutePath: The affected code is not reachable through the execution of the code, including non-anticipated states of the product.
- vulnerableCodeNotPresent: The product is not affected because the code underlying the vulnerability is not present in the product.

Licensing

Summary

The Licensing Profile defines a minimum set of license information to facilitate compliance with typical license use cases.

Description

The Licensing profile only contains the additional requirement that any Software Artifact must have a concludedLicense Relationship.

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

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

A declaredLicense identifies the license information actually found in the Software Artifact, for example as detected by use of automated tooling.

This field is not intended to capture license information obtained from an external source, such as a package's website. Such information can be included, as needed, in the concludedLicense field.

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

- for Packages:
- would include license info for the Package as a whole, found in the Package itself (e.g., LICENSE file, README file, metadata in the Package, etc.)
- would not include any license information that is not in the Package itself (e.g., license information from the project’s website or from a third party repository or website)
- for Files:
- would include license info found in the File itself (e.g., license header or notice, comments indicating the license, SPDX-License-Identifier expression)

SimpleLicensing

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/SimpleLicensing`

Name	SimpleLicensing

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/3.0.0/terms/SimpleLicensing/AnyLicenseInfo`

Name	AnyLicenseInfo
Instantiability	Abstract
SubclassOf	/Core/Element

Properties

Property	Type	minCount	maxCount

LicenseExpression

Summary

An SPDX Element containing an SPDX license expression string.

Description

A LicenseExpression enables the representation, in a single string, of a combination of one or more licenses, together with additions such as license exceptions.

The syntax for a LicenseExpression string is set forth in the SPDX License Expressions annex to the SPDX Specification. A LicenseExpression string is not valid if it does not conform to the grammar set forth in that annex.

The ExpandedLicensing profile can be used to represent the complete parsed license expression as a combination of license objects.

Metadata

`https://spdx.org/rdf/3.0.0/terms/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

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/3.0.0/terms/SimpleLicensing/SimpleLicensingText`

Name	SimpleLicensingText
Instantiability	Concrete
SubclassOf	/Core/Element

Properties

Property	Type	minCount	maxCount
licenseText	xsd:string	1	1

customIdToUri

Summary

Maps a LicenseRef or AdditionRef string for a Custom License or a Custom License Addition to its URI ID.

Description

Within a License Expression, references can be made to a Custom License or a Custom License Addition. The License Expression syntax dictates any reference starting with a "LicenseRef-" or "AdditionRef-" refers to license or addition text not found in the official SPDX License List. 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/3.0.0/terms/SimpleLicensing/customIdToUri`

Name	customIdToUri
Nature	ObjectProperty
Range	/Core/DictionaryEntry

Referenced

- /SimpleLicensing/LicenseExpression

licenseExpression

Summary

A string in the license expression format.

Description

A licenseExpression enables the representation, in a single string, of a combination of one or more licenses, together with additions such as license exceptions.

The syntax for a LicenseExpression string is set forth in the SPDX License Expressions annex to the SPDX Specification. A LicenseExpression string is not valid if it does not conform to the grammar set forth in that annex.

The ExpandedLicensing profile can be used to represent the complete parsed license expression as a combination of license objects.

Metadata

`https://spdx.org/rdf/3.0.0/terms/SimpleLicensing/licenseExpression`

Name	licenseExpression
Nature	DataProperty
Range	xsd:string

Referenced

- /SimpleLicensing/LicenseExpression

licenseListVersion

Summary

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

Description

Recognizing that licenses are added to the SPDX License List with each subsequent version, the intent is to provide consumers with the version of the SPDX License List used. This anticipates that in the future, license expression might have used a version of the SPDX License List that is older than the then current one. The specified version of the SPDX License List must include all listed licenses and exceptions referenced in the expression.

Metadata

`https://spdx.org/rdf/3.0.0/terms/SimpleLicensing/licenseListVersion`

Name	licenseListVersion
Nature	DataProperty
Range	/Core/SemVer

Referenced

- /SimpleLicensing/LicenseExpression

licenseText

Summary

Identifies the full text of a License or Addition.

Description

A licenseText contains the plain text of the License or Addition, without templating or other similar markup.

Users of the licenseText for a License can apply the SPDX Matching Guidelines when comparing it to another text for matching purposes.

Metadata

`https://spdx.org/rdf/3.0.0/terms/SimpleLicensing/licenseText`

Name	licenseText
Nature	DataProperty
Range	xsd:string

Referenced

- /ExpandedLicensing/License
- /SimpleLicensing/SimpleLicensingText

ExpandedLicensing

Summary

Fully expanded license expressions.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing`

Name	ExpandedLicensing

ConjunctiveLicenseSet

Summary

Portion of an AnyLicenseInfo representing a set of licensing information where all elements apply.

Description

A ConjunctiveLicenseSet indicates that *each* of its subsidiary AnyLicenseInfos apply. In other words, a ConjunctiveLicenseSet of two or more licenses represents a licensing situation where *all* of the specified licenses are to be complied with. It is represented in the SPDX License Expression Syntax by the `AND` operator.

It is syntactically correct to specify a ConjunctiveLicenseSet where the subsidiary AnyLicenseInfos may be "incompatible" according to a particular interpretation of the corresponding Licenses. The SPDX License Expression Syntax does not take into account interpretation of license texts, which is left to the consumer of SPDX data to determine for themselves.

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/ConjunctiveLicenseSet`

Name	ConjunctiveLicenseSet
Instantiability	Concrete
SubclassOf	/SimpleLicensing/AnyLicenseInfo

Properties

Property	Type	minCount	maxCount
member	/SimpleLicensing/AnyLicenseInfo	2	*

CustomLicense

Summary

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

Description

A CustomLicense represents a License that is not listed on the SPDX License List at https://spdx.org/licenses, and is therefore defined by an SPDX data creator.

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/CustomLicense`

Name	CustomLicense
Instantiability	Concrete
SubclassOf	License

Properties

Property	Type	minCount	maxCount

CustomLicenseAddition

Summary

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

Description

A CustomLicenseAddition represents an addition to a License that is not listed on the SPDX Exceptions List at https://spdx.org/licenses/exceptions-index.html, and is therefore defined by an SPDX data creator.

It is intended to represent additional language which is meant to be added to a License, but which is not itself a standalone License.

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/CustomLicenseAddition`

Name	CustomLicenseAddition
Instantiability	Concrete
SubclassOf	LicenseAddition

Properties

Property	Type	minCount	maxCount

DisjunctiveLicenseSet

Summary

Portion of an AnyLicenseInfo representing a set of licensing information where only one of the elements applies.

Description

A DisjunctiveLicenseSet indicates that *only one* of its subsidiary AnyLicenseInfos is required to apply. In other words, a DisjunctiveLicenseSet of two or more licenses represents a licensing situation where *only one* of the specified licenses are to be complied with. A consumer of SPDX data would typically understand this to permit the recipient of the licensed content to choose which of the corresponding license they would prefer to use. It is represented in the SPDX License Expression Syntax by the `OR` operator.

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/DisjunctiveLicenseSet`

Name	DisjunctiveLicenseSet
Instantiability	Concrete
SubclassOf	/SimpleLicensing/AnyLicenseInfo

Properties

Property	Type	minCount	maxCount
member	/SimpleLicensing/AnyLicenseInfo	2	*

ExtendableLicense

Summary

Abstract class representing a License or an OrLaterOperator.

Description

The WithAdditionOperator can have a License or an OrLaterOperator as the license property value. This class is used for the value.

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/ExtendableLicense`

Name	ExtendableLicense
Instantiability	Abstract
SubclassOf	/SimpleLicensing/AnyLicenseInfo

Properties

Property	Type	minCount	maxCount

IndividualLicensingInfo

Summary

A concrete subclass of AnyLicenseInfo used by Individuals in the ExpandedLicensing profile.

Description

Individuals, such as NoneLicense and NoAssertionLicense, need to reference a concrete subclass of AnyLicenseInfo.

This class provides the type used by the individuals.

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/IndividualLicensingInfo`

Name	IndividualLicensingInfo
Instantiability	Concrete
SubclassOf	/SimpleLicensing/AnyLicenseInfo

Properties

Property	Type	minCount	maxCount

License

Summary

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

Description

A License represents a license text, whether listed on the SPDX License List (ListedLicense) or defined by an SPDX data creator (CustomLicense).

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/License`

Name	License
Instantiability	Abstract
SubclassOf	ExtendableLicense

Properties

Property	Type	minCount	maxCount
/SimpleLicensing/licenseText	xsd:string	1	1
isDeprecatedLicenseId	xsd:boolean	0	1
isFsfLibre	xsd:boolean	0	1
isOsiApproved	xsd:boolean	0	1
licenseXml	xsd:string	0	1
obsoletedBy	xsd:string	0	1
seeAlso	xsd:anyURI	0	*
standardLicenseHeader	xsd:string	0	1
standardLicenseTemplate	xsd:string	0	1

LicenseAddition

Summary

Abstract class for additional text intended to be added to a License, but which is not itself a standalone License.

Description

A LicenseAddition represents text which is intended to be added to a License as additional text, but which is not itself intended to be a standalone License.

It may be an exception which is listed on the SPDX Exceptions List (ListedLicenseException), or may be any other additional text (as an exception or otherwise) which is defined by an SPDX data creator (CustomLicenseAddition).

Metadata

`https://spdx.org/rdf/3.0.0/terms/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

ListedLicense

Summary

A license that is listed on the SPDX License List.

Description

A ListedLicense represents a License that is listed on the SPDX License List at https://spdx.org/licenses.

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/ListedLicense`

Name	ListedLicense
Instantiability	Concrete
SubclassOf	License

Properties

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

ListedLicenseException

Summary

A license exception that is listed on the SPDX Exceptions list.

Description

A ListedLicenseException represents an exception to a License (in other words, an exception to a license condition or an additional permission beyond those granted in a License) which is listed on the SPDX Exceptions List at https://spdx.org/licenses/exceptions-index.html.

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/ListedLicenseException`

Name	ListedLicenseException
Instantiability	Concrete
SubclassOf	LicenseAddition

Properties

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

OrLaterOperator

Summary

Portion of an AnyLicenseInfo representing this version, or any later version, of the indicated License.

Description

An OrLaterOperator indicates that this portion of the AnyLicenseInfo represents either (1) the specified version of the corresponding License, or (2) any later version of that License. It is represented in the SPDX License Expression Syntax by the `+` operator.

It is context-dependent, and unspecified by SPDX, as to what constitutes a "later version" of any particular License. Some Licenses may not be versioned, or may not have clearly-defined ordering for versions. The consumer of SPDX data will need to determine for themselves what meaning to attribute to a "later version" operator for a particular License.

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/OrLaterOperator`

Name	OrLaterOperator
Instantiability	Concrete
SubclassOf	ExtendableLicense

Properties

Property	Type	minCount	maxCount
subjectLicense	License	1	1

WithAdditionOperator

Summary

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

Description

A WithAdditionOperator indicates that the designated License is subject to the designated LicenseAddition, which might be a license exception on the SPDX Exceptions List (ListedLicenseException) or may be other additional text (CustomLicenseAddition). It is represented in the SPDX License Expression Syntax by the `WITH` operator.

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/WithAdditionOperator`

Name	WithAdditionOperator
Instantiability	Concrete
SubclassOf	/SimpleLicensing/AnyLicenseInfo

Properties

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

additionText

Summary

Identifies the full text of a LicenseAddition.

Description

An additionText contains the plain text of the LicenseAddition, without templating or other similar markup.

Users of the additionText for a License can apply the SPDX Matching Guidelines when comparing it to another text for matching purposes.

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/additionText`

Name	additionText
Nature	DataProperty
Range	xsd:string

Referenced

- /ExpandedLicensing/LicenseAddition

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/3.0.0/terms/ExpandedLicensing/deprecatedVersion`

Name	deprecatedVersion
Nature	DataProperty
Range	xsd:string

Referenced

- /ExpandedLicensing/ListedLicense
- /ExpandedLicensing/ListedLicenseException

isDeprecatedAdditionId

Summary

Specifies whether an additional text identifier has been marked as deprecated.

Description

The isDeprecatedAdditionId property specifies whether an identifier for a LicenseAddition has been marked as deprecated. If the property is not defined, then it is presumed to be false (i.e., not deprecated).

If the LicenseAddition is included on the SPDX Exceptions List, then the `deprecatedVersion` property indicates on which version release of the Exceptions List it was first marked as deprecated.

"Deprecated" in this context refers to deprecating the use of the *identifier*, not the underlying license addition. In other words, even if a LicenseAddition's author or steward has stated that a particular LicenseAddition generally should not be used, that would *not* mean that the LicenseAddition's identifier is "deprecated." Rather, a LicenseAddition operator is typically marked as "deprecated" when it is determined that use of another identifier is preferable.

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/isDeprecatedAdditionId`

Name	isDeprecatedAdditionId
Nature	DataProperty
Range	xsd:boolean

Referenced

- /ExpandedLicensing/LicenseAddition

isDeprecatedLicenseId

Summary

Specifies whether a license or additional text identifier has been marked as deprecated.

Description

The isDeprecatedLicenseId property specifies whether an identifier for a License or LicenseAddition has been marked as deprecated. If the property is not defined, then it is presumed to be false (i.e., not deprecated).

If the License or LicenseAddition is included on the SPDX License List, then the `deprecatedVersion` property indicates on which version release of the License List it was first marked as deprecated.

"Deprecated" in this context refers to deprecating the use of the *identifier*, not the underlying license. In other words, even if a License's author or steward has stated that a particular License generally should not be used, that would *not* mean that the License's identifier is "deprecated." Rather, a License or LicenseAddition operator is typically marked as "deprecated" when it is determined that use of another identifier is preferable.

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/isDeprecatedLicenseId`

Name	isDeprecatedLicenseId
Nature	DataProperty
Range	xsd:boolean

Referenced

- /ExpandedLicensing/License

isFsfLibre

Summary

Specifies whether the License is listed as free by the Free Software Foundation (FSF).

Description

isFsfLibre specifies whether the Free Software Foundation FSF has listed this License as "free" in their commentary on licenses, located at the time of this writing at https://www.gnu.org/licenses/license-list.en.html.

A value of "true" indicates that the license is in the list of licenses that FSF publishes as libre.

A value of "false" indicates that the license is explicitly not in the corresponding list of FSF libre licenses (e.g., FSF has the license on a non-free list).

If the isFsfLibre field is not specified, the SPDX data creator makes no assertions about whether the License is listed in the FSF's commentary.

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/isFsfLibre`

Name	isFsfLibre
Nature	DataProperty
Range	xsd:boolean

Referenced

- /ExpandedLicensing/License

isOsiApproved

Summary

Specifies whether the License is listed as approved by the Open Source Initiative (OSI).

Description

isOsiApproved specifies whether the Open Source Initiative (OSI) has listed this License as "approved" in their list of OSI Approved Licenses, located at the time of this writing at https://opensource.org/licenses/.

A value of "true" indicates that the license is in the list of licenses that OSI publishes as approved.

A value of "false" indicates that the license is explicitly not in the corresponding list of OSI licenses (e.g., OSI has stated publicly that a license is not approved).

If the isOsiApproved field is not specified, the SPDX data creator makes no assertions about whether the License is approved by the OSI.

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/isOsiApproved`

Name	isOsiApproved
Nature	DataProperty
Range	xsd:boolean

Referenced

- /ExpandedLicensing/License

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/3.0.0/terms/ExpandedLicensing/licenseXml`

Name	licenseXml
Nature	DataProperty
Range	xsd:string

Referenced

- /ExpandedLicensing/License
- /ExpandedLicensing/LicenseAddition

listVersionAdded

Summary

Specifies the SPDX License List version in which this ListedLicense or ListedLicenseException identifier was first added.

Description

A listVersionAdded for a ListedLicense or ListedLicenseException on the SPDX License List specifies which version release of the License List was the first one in which it was included.

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/listVersionAdded`

Name	listVersionAdded
Nature	DataProperty
Range	xsd:string

Referenced

- /ExpandedLicensing/ListedLicense
- /ExpandedLicensing/ListedLicenseException

member

Summary

A license expression participating in a license set.

Description

A member is a license expression participating in a conjunctive (of type ConjunctiveLicenseSet) or a disjunctive (of type DisjunctiveLicenseSet) license set.

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/member`

Name	member
Nature	ObjectProperty
Range	/SimpleLicensing/AnyLicenseInfo

Referenced

- /ExpandedLicensing/ConjunctiveLicenseSet
- /ExpandedLicensing/DisjunctiveLicenseSet

obsoletedBy

Summary

Specifies the licenseId that is preferred to be used in place of a deprecated License or LicenseAddition.

Description

An obsoletedBy value for a deprecated License or LicenseAddition specifies the licenseId of the replacement License or LicenseAddition that is preferred to be used in its place. It should use the same format as specified for a licenseId.

The License's or LicenseAddition's comment value may include more information about the reason why the licenseId specified in the obsoletedBy value is preferred.

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/obsoletedBy`

Name	obsoletedBy
Nature	DataProperty
Range	xsd:string

Referenced

- /ExpandedLicensing/License
- /ExpandedLicensing/LicenseAddition

seeAlso

Summary

Contains a URL where the License or LicenseAddition can be found in use.

Description

A seeAlso defines a cross-reference with a URL where the License or LicenseAddition can be found in use by one or a few projects.

If applicable, it should include a URL where the license text is posted by the license steward, particularly if the license steward has made available a "canonical" primary URL for the license text.

If the license is OSI approved, a seeAlso should be included with the URL for the license's listing on the OSI website.

The seeAlso URL may refer to a previously-available URL for the License or LicenseAddition which is no longer active.

Where applicable, the seeAlso URL should include the license text in its native language. seeAlso URLs to English or other translations may be included where multiple, equivalent official translations exist.

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/seeAlso`

Name	seeAlso
Nature	DataProperty
Range	xsd:anyURI

Referenced

- /ExpandedLicensing/License
- /ExpandedLicensing/LicenseAddition

standardAdditionTemplate

Summary

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

Description

A standardAdditionTemplate contains a license addition template which describes sections of the LicenseAddition text which can be varied. See the Legacy Text Template format section of the SPDX specification for format information.

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/standardAdditionTemplate`

Name	standardAdditionTemplate
Nature	DataProperty
Range	xsd:string

Referenced

- /ExpandedLicensing/LicenseAddition

standardLicenseHeader

Summary

Provides a License author's preferred text to indicate that a file is covered by the License.

Description

A standardLicenseHeader contains the plain text of the License author's preferred wording to be used, typically in a source code file's header comments or similar location, to indicate that the file is subject to the specified License.

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/standardLicenseHeader`

Name	standardLicenseHeader
Nature	DataProperty
Range	xsd:string

Referenced

- /ExpandedLicensing/License

standardLicenseTemplate

Summary

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

Description

A standardLicenseTemplate contains a license template which describes sections of the License text which can be varied. See the Legacy Text Template format section of the SPDX specification for format information.

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/standardLicenseTemplate`

Name	standardLicenseTemplate
Nature	DataProperty
Range	xsd:string

Referenced

- /ExpandedLicensing/License

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/3.0.0/terms/ExpandedLicensing/subjectAddition`

Name	subjectAddition
Nature	ObjectProperty
Range	LicenseAddition

Referenced

- /ExpandedLicensing/WithAdditionOperator

subjectExtendableLicense

Summary

A License participating in a 'with addition' model.

Description

A subjectExtendableLicense is a License which is subject to a 'with additional text' effect (WithAdditionOperator).

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/subjectExtendableLicense`

Name	subjectExtendableLicense
Nature	ObjectProperty
Range	ExtendableLicense

Referenced

- /ExpandedLicensing/WithAdditionOperator

subjectLicense

Summary

A License participating in an 'or later' model.

Description

A subjectLicense is a License which is subject an 'or later' effect (OrLaterOperator).

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/subjectLicense`

Name	subjectLicense
Nature	ObjectProperty
Range	License

Referenced

- /ExpandedLicensing/OrLaterOperator

NoAssertionLicense

Summary

An Individual Value for License when no assertion can be made about its actual value.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/NoAssertionLicense`

Name	NoAssertionLicense
Type	IndividualLicensingInfo
IRI	`https://spdx.org/rdf/3.0.0/terms/Licensing/NoAssertion`

NoneLicense

Summary

An Individual Value for License where the SPDX data creator determines that no license is present.

Description

NoneLicense should be used if the SPDX creator determines there is no license available for this Artifact.

Metadata

`https://spdx.org/rdf/3.0.0/terms/ExpandedLicensing/NoneLicense`

Name	NoneLicense
Type	IndividualLicensingInfo
IRI	`https://spdx.org/rdf/3.0.0/terms/Licensing/None`

Dataset

Summary

The Dataset Profile provides additional metadata, based on Software Profile, that is useful for datasets.

Description

The Dataset namespace defines concepts related to dataset, including its preparation process, its characteristics, and its access methods.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Dataset`

Name	Dataset

DatasetPackage

Summary

Specifies a data package and its associated information.

Description

Metadata information that can be added to a dataset that may be used in a software or to train/test an AI package.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Dataset/DatasetPackage`

Name	DatasetPackage
Instantiability	Concrete
SubclassOf	/Software/Package

Properties

Property	Type	minCount	maxCount
anonymizationMethodUsed	xsd:string	0	*
confidentialityLevel	ConfidentialityLevelType	0	1
dataCollectionProcess	xsd:string	0	1
dataPreprocessing	xsd:string	0	*
datasetAvailability	DatasetAvailabilityType	0	1
datasetNoise	xsd:string	0	1
datasetSize	xsd:nonNegativeInteger	0	1
datasetType	DatasetType	1	*
datasetUpdateMechanism	xsd:string	0	1
hasSensitivePersonalInformation	/Core/PresenceType	0	1
intendedUse	xsd:string	0	1
knownBias	xsd:string	0	*
sensor	/Core/DictionaryEntry	0	*

anonymizationMethodUsed

Summary

Describes the anonymization methods used.

Description

A free-form text that describes the methods used to anonymize the dataset (of fields in the dataset).

Metadata

`https://spdx.org/rdf/3.0.0/terms/Dataset/anonymizationMethodUsed`

Name	anonymizationMethodUsed
Nature	DataProperty
Range	xsd:string

Referenced

- /Dataset/DatasetPackage

confidentialityLevel

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Dataset/confidentialityLevel`

Name	confidentialityLevel
Nature	ObjectProperty
Range	ConfidentialityLevelType

Referenced

- /Dataset/DatasetPackage

dataCollectionProcess

Summary

Describes how the dataset was collected.

Description

A free-form text that describes how a dataset was collected.

Examples include the sources from which a dataset was scrapped and the interview protocol that was used for data collection.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Dataset/dataCollectionProcess`

Name	dataCollectionProcess
Nature	DataProperty
Range	xsd:string

Referenced

- /Dataset/DatasetPackage

dataPreprocessing

Summary

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

Description

A free-form text that describes the various preprocessing steps that were applied to the raw data to create the dataset.

Examples include standardization, normalization, deduplication, tokenization, and removal of tokens.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Dataset/dataPreprocessing`

Name	dataPreprocessing
Nature	DataProperty
Range	xsd:string

Referenced

- /Dataset/DatasetPackage

datasetAvailability

Summary

The field describes the availability of a dataset.

Description

Some datasets are publicly available and can be downloaded directly. Others are only accessible behind a clickthrough, or after filling a registration form. This field will describe the dataset availability from that perspective.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Dataset/datasetAvailability`

Name	datasetAvailability
Nature	ObjectProperty
Range	DatasetAvailabilityType

Referenced

- /Dataset/DatasetPackage

datasetNoise

Summary

Describes potentially noisy elements of the dataset.

Description

Describes what kinds of noises a dataset might encompass.

The free-form text specifies fields or samples that might be noisy.

Alternatively, it can also be used to describe various noises that could impact the whole dataset.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Dataset/datasetNoise`

Name	datasetNoise
Nature	DataProperty
Range	xsd:string

Referenced

- /Dataset/DatasetPackage

datasetSize

Summary

Captures the size of the dataset.

Description

Captures how large a dataset is.

The size is to be measured in bytes.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Dataset/datasetSize`

Name	datasetSize
Nature	DataProperty
Range	xsd:nonNegativeInteger

Referenced

- /Dataset/DatasetPackage

datasetType

Summary

Describes the type of the given dataset.

Description

Describes the datatype contained in the dataset.

For example, a dataset can be an image dataset for computer vision applications, a text dataset such as the contents of a book or Wikipedia article, or sometimes a multimodal dataset that contains multiple types of data.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Dataset/datasetType`

Name	datasetType
Nature	ObjectProperty
Range	DatasetType

Referenced

- /Dataset/DatasetPackage

datasetUpdateMechanism

Summary

Describes a mechanism to update the dataset.

Description

A free-form text that describes a mechanism to update the dataset.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Dataset/datasetUpdateMechanism`

Name	datasetUpdateMechanism
Nature	DataProperty
Range	xsd:string

Referenced

- /Dataset/DatasetPackage

hasSensitivePersonalInformation

Summary

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

Description

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

Related: `useSensitivePersonalInformation` in `/AI/AIPackage`

Metadata

`https://spdx.org/rdf/3.0.0/terms/Dataset/hasSensitivePersonalInformation`

Name	hasSensitivePersonalInformation
Nature	ObjectProperty
Range	/Core/PresenceType

Referenced

- /Dataset/DatasetPackage

intendedUse

Summary

Describes what the given dataset should be used for.

Description

A free-form text that describes what the given dataset should be used for.

Some datasets are collected to be used only for particular purposes.

For example, medical data collected from a specific demography might only be applicable for training machine learning models to make predictions for that demography. In such a case, the intendedUse field would capture this information. Similarly, if a dataset is collected for building a facial recognition model, the intendedUse field would specify that.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Dataset/intendedUse`

Name	intendedUse
Nature	DataProperty
Range	xsd:string

Referenced

- /Dataset/DatasetPackage

knownBias

Summary

Records the biases that the dataset is known to encompass.

Description

A free-form text that describes the different biases that the dataset encompasses.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Dataset/knownBias`

Name	knownBias
Nature	DataProperty
Range	xsd:string

Referenced

- /Dataset/DatasetPackage

sensor

Summary

Describes a sensor used for collecting the data.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Dataset/sensor`

Name	sensor
Nature	ObjectProperty
Range	/Core/DictionaryEntry

Referenced

- /Dataset/DatasetPackage

ConfidentialityLevelType

Summary

Categories of confidentiality level.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Dataset/ConfidentialityLevelType`

Name	ConfidentialityLevelType

Entries

- amber: Data points in the dataset can be shared only with specific organizations and their clients on a need to know basis.
- clear: Dataset may be distributed freely, without restriction.
- green: Dataset can be shared within a community of peers and partners.
- red: Data points in the dataset are highly confidential and can only be shared with named recipients.

DatasetAvailabilityType

Summary

Availability of dataset

Description

Describes the possible types of availability of a dataset, indicating whether the dataset can be directly downloaded, can be assembled using a script for scraping the data, is only available after a clickthrough or a registration form.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Dataset/DatasetAvailabilityType`

Name	DatasetAvailabilityType

Entries

- clickthrough: the dataset is not publicly available and can only be accessed after affirmatively accepting terms on a clickthrough webpage.
- directDownload: the dataset is publicly available and can be downloaded directly.
- query: the dataset is publicly available, but not all at once, and can only be accessed through queries which return parts of the dataset.
- registration: the dataset is not publicly available and an email registration is required before accessing the dataset, although without an affirmative acceptance of terms.
- scrapingScript: the dataset provider is not making available the underlying data and the dataset must be reassembled, typically using the provided script for scraping the data.

DatasetType

Summary

Enumeration of dataset types.

Description

Describes the different structures of data within a given dataset. A dataset can have multiple types of data, or even a single type of data but still match multiple types, for example sensor data could also be timeseries or labeled image data could also be considered categorical.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Dataset/DatasetType`

Name	DatasetType

Entries

- audio: data is audio based, such as a collection of music from the 80s.
- categorical: data that is classified into a discrete number of categories, such as the eye color of a population of people.
- graph: data is in the form of a graph where entries are somehow related to each other through edges, such a social network of friends.
- image: data is a collection of images such as pictures of animals.
- noAssertion: data type is not known.
- numeric: data consists only of numeric entries.
- other: data is of a type not included in this list.
- sensor: data is recorded from a physical sensor, such as a thermometer reading or biometric device.
- structured: data is stored in tabular format or retrieved from a relational database.
- syntactic: data describes the syntax or semantics of a language or text, such as a parse tree used for natural language processing.
- text: data consists of unstructured text, such as a book, Wikipedia article (without images), or transcript.

AI

Summary

The AI Profile is designed to provide a standardized way of documenting and sharing information about AI software packages (i.e. systems).

Description

The AI namespace defines a set of concepts and data elements related to AI system and model artifacts. These artifacts are the tangible outputs of the AI development process, such as software packages, models, and datasets.

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI`

Name	AI

AIPackage

Summary

Specifies an AI package and its associated information.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/AIPackage`

Name	AIPackage
Instantiability	Concrete
SubclassOf	/Software/Package

Properties

Property	Type	minCount	maxCount
autonomyType	/Core/PresenceType	0	1
domain	xsd:string	0	*
energyConsumption	EnergyConsumption	0	1
hyperparameter	/Core/DictionaryEntry	0	*
informationAboutApplication	xsd:string	0	1
informationAboutTraining	xsd:string	0	1
limitation	xsd:string	0	1
metric	/Core/DictionaryEntry	0	*
metricDecisionThreshold	/Core/DictionaryEntry	0	*
modelDataPreprocessing	xsd:string	0	*
modelExplainability	xsd:string	0	*
safetyRiskAssessment	SafetyRiskAssessmentType	0	1
standardCompliance	xsd:string	0	*
typeOfModel	xsd:string	0	*
useSensitivePersonalInformation	/Core/PresenceType	0	1

EnergyConsumption

Summary

The class that contains properties to describe energy consumption incurred by an AI model in different stages of its lifecycle.

Description

The class used for denoting the training energy consumption, inference energy consumption and fine tuning energy consumption of the AI model(s) used in an AI system.

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/EnergyConsumption`

Name	EnergyConsumption
Instantiability	Concrete

Properties

Property	Type	minCount	maxCount
finetuningEnergyConsumption	EnergyConsumptionDescription	0	*
inferenceEnergyConsumption	EnergyConsumptionDescription	0	*
trainingEnergyConsumption	EnergyConsumptionDescription	0	*

EnergyConsumptionDescription

Summary

The class that helps note down the quantity of energy consumption and the unit used for measurement.

Description

This class is designed to store energy consumption data, including the quantity and the unit of measurement.

The energyQuantity property stores the amount of energy consumed, and the energyUnit property stores the unit used for measurement.

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/EnergyConsumptionDescription`

Name	EnergyConsumptionDescription
Instantiability	Concrete

Properties

Property	Type	minCount	maxCount
energyQuantity	xsd:decimal	1	1
energyUnit	EnergyUnitType	1	1

autonomyType

Summary

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

Description

Indicates if a human is involved in any of the decisions of the AI system or if that system is fully automatic.

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/autonomyType`

Name	autonomyType
Nature	ObjectProperty
Range	/Core/PresenceType

Referenced

- /AI/AIPackage

domain

Summary

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

Description

A free-form text that describes the domain where the AI model contained in the AI software can be expected to operate successfully. Examples include computer vision, natural language processing, etc.

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/domain`

Name	domain
Nature	DataProperty
Range	xsd:string

Referenced

- /AI/AIPackage

energyConsumption

Summary

Indicates the amount of energy consumed to train the AI model.

Description

A free-form text captures known or estimated energy consumption for the training of the AI model.

In case not known, the estimation could be based on information about computational resources used (e.g. number of floating point operations – FLOPs), training time, type and quantity of processing units, and other relevant details related to the training.

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/energyConsumption`

Name	energyConsumption
Nature	ObjectProperty
Range	EnergyConsumption

Referenced

- /AI/AIPackage

energyQuantity

Summary

Represents the energy quantity.

Description

Provides the quantity information of the energy.

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/energyQuantity`

Name	energyQuantity
Nature	DataProperty
Range	xsd:decimal

Referenced

- /AI/EnergyConsumptionDescription

energyUnit

Summary

Specifies the unit in which energy is measured.

Description

Provides the unit information of the energy.

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/energyUnit`

Name	energyUnit
Nature	ObjectProperty
Range	EnergyUnitType

Referenced

- /AI/EnergyConsumptionDescription

finetuningEnergyConsumption

Summary

Specifies the amount of energy consumed when finetuning the AI model that is being used in the AI system.

Description

The field specifies the amount of energy consumed when finetuning the AI model that is being used in the AI system.

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/finetuningEnergyConsumption`

Name	finetuningEnergyConsumption
Nature	ObjectProperty
Range	EnergyConsumptionDescription

Referenced

- /AI/EnergyConsumption

hyperparameter

Summary

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

Description

Records a hyperparameter value.

Hyperparameters are settings defined before the training process that control the learning algorithm's behavior. They differ from model parameters, which are learned from the data during training. Developers typically set hyperparameters manually or through a process of hyperparameter tuning (also known as trial and error).

Examples of hyperparameters include learning rate, batch size, and the number of layers in a neural network.

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/hyperparameter`

Name	hyperparameter
Nature	ObjectProperty
Range	/Core/DictionaryEntry

Referenced

- /AI/AIPackage

inferenceEnergyConsumption

Summary

Specifies the amount of energy consumed during inference time by an AI model that is being used in the AI system.

Description

The field specifies the amount of energy consumed during inference time by an AI model that is being used in the AI system.

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/inferenceEnergyConsumption`

Name	inferenceEnergyConsumption
Nature	ObjectProperty
Range	EnergyConsumptionDescription

Referenced

- /AI/EnergyConsumption

informationAboutApplication

Summary

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

Description

A free-form text description of how the AI model is used within the software. It should include any relevant information, such as pre-processing steps, third-party APIs, and other pertinent details.

It can also include:

- Functionality provided by the AI model within the software application, including: any specific tasks or decisions it is designed to perform; any pre-processing steps that are applied to the input data before it is fed into the AI model for inference, such as data cleaning, normalization, or feature extraction; and any third-party APIs or services that are used in conjunction with the AI model, such as data sources, cloud services, or other AI models.
- Description of any dependencies or requirements needed to run the AI model within the software application, including: specific hardware, software libraries, and operating systems.

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/informationAboutApplication`

Name	informationAboutApplication
Nature	DataProperty
Range	xsd:string

Referenced

- /AI/AIPackage

informationAboutTraining

Summary

Describes relevant information about different steps of the training process.

Description

A detailed explanation of the training process, including the specific techniques, algorithms, and methods employed.

Examples include:

- training data used to train the AI model, along with any relevant details about its source, quality, and pre-processing steps;
- specific training algorithms employed, including stochastic gradient descent, backpropagation, and reinforcement learning.
- specific training techniques used to improve the performance or accuracy of the AI model, such as transfer learning, fine-tuning, or active learning; and
- any evaluation metrics used to assess the performance of the AI model during the training process, including accuracy, precision, recall, and F1 score.

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/informationAboutTraining`

Name	informationAboutTraining
Nature	DataProperty
Range	xsd:string

Referenced

- /AI/AIPackage

limitation

Summary

Captures a limitation of the AI software.

Description

A free-form text that captures a limitation of the AI package (or of the AI models present in the AI package). Note that this is not guaranteed to be exhaustive. For instance, a limitation might be that the AI package cannot be used on datasets from a certain demography.

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/limitation`

Name	limitation
Nature	DataProperty
Range	xsd:string

Referenced

- /AI/AIPackage

metric

Summary

Records the measurement of prediction quality of the AI model.

Description

Records the measurement with which the AI model was evaluated. This makes statements about the prediction quality including uncertainty, accuracy, characteristics of the tested population, quality, fairness, explainability, robustness etc.

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/metric`

Name	metric
Nature	ObjectProperty
Range	/Core/DictionaryEntry

Referenced

- /AI/AIPackage

metricDecisionThreshold

Summary

Captures the threshold that was used for computation of a metric described in the metric field.

Description

Each metric might be computed based on a decision threshold. For instance, precision or recall is typically computed by checking if the probability of the outcome is larger than 0.5. Each decision threshold should match with a metric field defined in the AI package.

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/metricDecisionThreshold`

Name	metricDecisionThreshold
Nature	ObjectProperty
Range	/Core/DictionaryEntry

Referenced

- /AI/AIPackage

modelDataPreprocessing

Summary

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

Description

A free-form text that describes the preprocessing steps applied to the training data before training of the model(s) contained in the AI software.

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/modelDataPreprocessing`

Name	modelDataPreprocessing
Nature	DataProperty
Range	xsd:string

Referenced

- /AI/AIPackage

modelExplainability

Summary

Describes methods that can be used to explain the model.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/modelExplainability`

Name	modelExplainability
Nature	DataProperty
Range	xsd:string

Referenced

- /AI/AIPackage

safetyRiskAssessment

Summary

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

Description

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

Using categorization according to the EU general risk assessment methodology. 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/3.0.0/terms/AI/safetyRiskAssessment`

Name	safetyRiskAssessment
Nature	ObjectProperty
Range	SafetyRiskAssessmentType

Referenced

- /AI/AIPackage

standardCompliance

Summary

Captures a standard that is being complied with.

Description

A free-form text that captures a standard that the AI software complies with.

This includes both published and unpublished standards, such as those developed by ISO, IEEE, and ETSI.

The standard may, but is not necessarily required to, satisfy a legal or regulatory requirement.

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/standardCompliance`

Name	standardCompliance
Nature	DataProperty
Range	xsd:string

Referenced

- /AI/AIPackage

trainingEnergyConsumption

Summary

Specifies the amount of energy consumed when training the AI model that is being used in the AI system.

Description

The field specifies the amount of energy consumed when training the AI model that is being used in the AI system.

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/trainingEnergyConsumption`

Name	trainingEnergyConsumption
Nature	ObjectProperty
Range	EnergyConsumptionDescription

Referenced

- /AI/EnergyConsumption

typeOfModel

Summary

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

Description

A free-form text that records the type of the AI model(s) used in the software. For instance, if it is a supervised model, unsupervised model, reinforcement learning model or a combination of those.

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/typeOfModel`

Name	typeOfModel
Nature	DataProperty
Range	xsd:string

Referenced

- /AI/AIPackage

useSensitivePersonalInformation

Summary

Records if sensitive personal information is used during model training or could be used during the inference.

Description

Notes if sensitive personal information is used in the training or inference of the AI models. This might include biometric data, addresses or other data that can be used to infer a person's identity.

Related: `hasSensitivePersonalInformation` in `/Dataset/DatasetPackage`

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/useSensitivePersonalInformation`

Name	useSensitivePersonalInformation
Nature	ObjectProperty
Range	/Core/PresenceType

Referenced

- /AI/AIPackage

EnergyUnitType

Summary

Specifies the unit of energy consumption.

Description

List the different acceptable units for measuring energy consumption.

If the unit in which the energy consumption has been recorded is not listed here, please select "other".

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/EnergyUnitType`

Name	EnergyUnitType

Entries

- kilowattHour: Kilowatt-hour.
- megajoule: Megajoule.
- other: Any other units of energy measurement.

SafetyRiskAssessmentType

Summary

Specifies the safety risk level.

Description

Lists the different general safety risk levels that can be used to describe the general safety risk of an AI system.

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/AI/SafetyRiskAssessmentType`

Name	SafetyRiskAssessmentType

Entries

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

Build

Summary

The Build Profile defines the set of information required to describe an instance of a Software Build.

Description

A Software Build is defined here as the act of converting software inputs into software artifacts using software build tools. Inputs can include source code, config files, artifacts that are build environments, and build tools. Outputs can include intermediate artifacts to other build inputs or the final artifacts.

The Build profile provides a subclass of Element called Build. It also provides a minimum set of required Relationship Types from the Core profile:

- hasInputs: Describes the relationship from the Build element to its inputs.
- hasOutputs: Describes the relationship from the Build element to its outputs.
- invokedBy: Describes the relationship from the Build element to the Agent that invoked it.

In addition, the following Relationship Types may be used to describe a Build.

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

All relationships in the Build Profile are scoped to the "build" LifecycleScopeType period.

The `hasInputs` relationship can be applied to a config file or a build tool if the nature of these inputs are not known at the creation of an SPDX document.

Build

Summary

Class that describes a build instance of software/artifacts.

Description

A build is a representation of the process in which a piece of software or artifact is built. It encapsulates information related to a build process and provides an element from which relationships can be created to describe the build's inputs, outputs, and related entities (e.g. builders, identities, etc.).

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

ExternalIdentifier of type "urlScheme" may be used to identify build logs. In this case, the comment of the ExternalIdentifier should be "LogReference".

Note that buildStartTime and buildEndTime are optional, and may be omitted to simplify creating reproducible builds.

Metadata

`https://spdx.org/rdf/3.0.0/terms/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	*

buildEndTime

Summary

Property that describes the time at which a build stops.

Description

buildEndTime describes the time at which a build stops or finishes. This value is typically recorded by the builder.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Build/buildEndTime`

Name	buildEndTime
Nature	DataProperty
Range	/Core/DateTime

Referenced

- /Build/Build

buildId

Summary

A buildId is a locally unique identifier used by a builder to identify a unique instance of a build produced by it.

Description

A buildId is a locally unique identifier to identify a unique instance of a build. This identifier differs based on build toolchain, platform, or naming convention used by an organization or standard.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Build/buildId`

Name	buildId
Nature	DataProperty
Range	xsd:string

Referenced

- /Build/Build

buildStartTime

Summary

Property describing the start time of a build.

Description

buildStartTime is the time at which a build is triggered. The builder typically records this value.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Build/buildStartTime`

Name	buildStartTime
Nature	DataProperty
Range	/Core/DateTime

Referenced

- /Build/Build

buildType

Summary

A buildType is a hint that is used to indicate the toolchain, platform, or infrastructure that the build was invoked on.

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Build/buildType`

Name	buildType
Nature	DataProperty
Range	xsd:anyURI

Referenced

- /Build/Build

configSourceDigest

Summary

Property that describes the digest of the build configuration file used to invoke a build.

Description

configSourceDigest is the checksum of the build configuration file used by a builder to execute a build. This Property uses the Core model's Hash class.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Build/configSourceDigest`

Name	configSourceDigest
Nature	ObjectProperty
Range	/Core/Hash

Referenced

- /Build/Build

configSourceEntrypoint

Summary

Property describes the invocation entrypoint of a build.

Description

A build entrypoint is the invoked executable of a build which always runs when the build is triggered. For example, when a build is triggered by running a shell script, the entrypoint is `script.sh`. In terms of a declared build, the entrypoint is the position in a configuration file or a build declaration which is always run when the build is triggered. For example, in the following configuration file, the entrypoint of the build is `publish`.

```
name: Publish packages to PyPI

on:
create:
tags: "*"

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

...
```

Metadata

`https://spdx.org/rdf/3.0.0/terms/Build/configSourceEntrypoint`

Name	configSourceEntrypoint
Nature	DataProperty
Range	xsd:string

Referenced

- /Build/Build

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/3.0.0/terms/Build/configSourceUri`

Name	configSourceUri
Nature	DataProperty
Range	xsd:anyURI

Referenced

- /Build/Build

environment

Summary

Property describing the session in which a build is invoked.

Description

environment is a map of environment variables and values that are set during a build session. This is different from the parameters property in that it describes the environment variables set before a build is invoked rather than the variables provided to the builder.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Build/environment`

Name	environment
Nature	ObjectProperty
Range	/Core/DictionaryEntry

Referenced

- /Build/Build

parameters

Summary

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

Description

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Build/parameters`

Name	parameters
Nature	ObjectProperty
Range	/Core/DictionaryEntry

Referenced

- /Build/Build

Lite

Summary

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

Description

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

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

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

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

Metadata

`https://spdx.org/rdf/3.0.0/terms/Lite`

Name	Lite

Extension

Summary

Everything having to do with SPDX extensions.

Description

The Extension namespace defines the abstract Extension class serving as the base for all defined extension subclasses.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Extension`

Name	Extension

CdxPropertiesExtension

Summary

A type of extension consisting of a list of name value pairs.

Description

This extension provides a more structured extension using a name-value approach. Unlike key-value stores, cdxProperties support duplicate names, each potentially having different values. This is intended to be compatible with the CycloneDX property `properties`.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Extension/CdxPropertiesExtension`

Name	CdxPropertiesExtension
Instantiability	Concrete
SubclassOf	Extension

Properties

Property	Type	minCount	maxCount
cdxProperty	CdxPropertyEntry	1	*

CdxPropertyEntry

Summary

A property name with an associated value.

Description

Each CdxPropertyEntry contains a name-value pair which maps the name to its associated value. Unlike key-value stores, cdxProperties support duplicate names, each potentially having different values. This class can be used to implement CycloneDX compatible properties.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Extension/CdxPropertyEntry`

Name	CdxPropertyEntry
Instantiability	Concrete

Properties

Property	Type	minCount	maxCount
cdxPropName	xsd:string	1	1
cdxPropValue	xsd:string	0	1

Extension

Summary

A characterization of some aspect of an Element that is associated with the Element in a generalized fashion.

Description

An Extension is a characterization of some aspect of an Element that is associated with the Element in a generalized fashion.

Rather than being associated with a particular Element through the typical use of a purpose-specific object property an Extension is associated with the Element it characterizes using a single common generalized object property.

This approach serves multiple purposes:

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

cdxPropName

Summary

A name used in a CdxExtension name-value pair.

Description

A cdxPropName is used in a CdxExtension name-value pair. Unlike key-value stores, cdxProperties support duplicate names, each potentially having different values.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Extension/cdxPropName`

Name	cdxPropName
Nature	DataProperty
Range	xsd:string

Referenced

- /Extension/CdxPropertyEntry

cdxPropValue

Summary

A value used in a CdxExtension name-value pair.

Description

A cdxPropValue is used in a CdxExtension name-value pair. Unlike key-value stores, cdxProperties support duplicate names, each potentially having different values.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Extension/cdxPropValue`

Name	cdxPropValue
Nature	DataProperty
Range	xsd:string

Referenced

- /Extension/CdxPropertyEntry

cdxProperty

Summary

Provides a map of a property names to a values.

Description

This field provides a mapping of a name to a value. This is intended to be compatible with the CycloneDX property "properties". Unlike key-value stores, properties support duplicate names, each potentially having different values.

Metadata

`https://spdx.org/rdf/3.0.0/terms/Extension/cdxProperty`

Name	cdxProperty
Nature	ObjectProperty
Range	CdxPropertyEntry

Referenced

- /Extension/CdxPropertiesExtension

Annex A: Differences from previous editions (Informative)

A.1 Differences between V3.0 and V2.3

Structural Differences

These are the most significant breaking changes requiring a change in logic to handle a different model or structure for the information. Each structural difference will describe the change, describe an approach to translate from 2.3 to 3.0, and provide a rationale for the change.

External Document Reference

Description of Change

The purpose of the SPDX 2.3 structure “ExternalDocumentRef” is now covered by two separate structures:

- NamespaceMap which maps short identifiers used in serializations to full namespace URI’s to support terseness in serialization of element identifiers
- ExternalMap which maps an element identifier for an element defined externally to verification and location information

The externalDocumentRef property on the SpdxDocument has been replaced by import property and namespace property.

Another change is the SPDX document checksum field has been replaced with a “verifiedUsing” property on the ElementCollection. The “verifiedUsing” which has 0 or more “IntegrityMethod” which should be the checksum of the SPDX document.

Annex B: Getting started writing SPDX 3 (Informative)

(a.k.a My First SPDX File)

This guide is designed to walk you through the concepts behind an SPDX document, by walking through writing one by hand. While it is possible to write all your SPDX documents by hand, we would recommend looking at the various language bindings that are available for crafting more complex documents. Nevertheless, walking through an example of a hand written document can be instructive into how SPDX documents work to better understand concepts that are at play, even when using language bindings.

All of the provided fragments listed here are intended to be used to construct a complete a valid SPDX JSON document when concatenated together

If you do would like to construct the complete example from this Markdown file, use the following command:

```
cat getting-started.md | awk '/^```json/, $0=="```" {if ($0 !~ /^```.*/ ) print}'
```

Please note that all descriptions of properties, classes, etc. are non-normative; that is they are intended to help you understand what is going on in simpler language, but are not necessarily complete. Links to the full official documentation are provided where possible.

The Preamble

All documents need to start somewhere, and SPDX documents are no exception.

The root of all SPDX documents will be a JSON object, so start with that:

```
{
```

Next, we need to identify that the document is an SPDX 3 JSON-LD document, which is done with:

Annex C: RDF object model and identifier syntax (Normative)

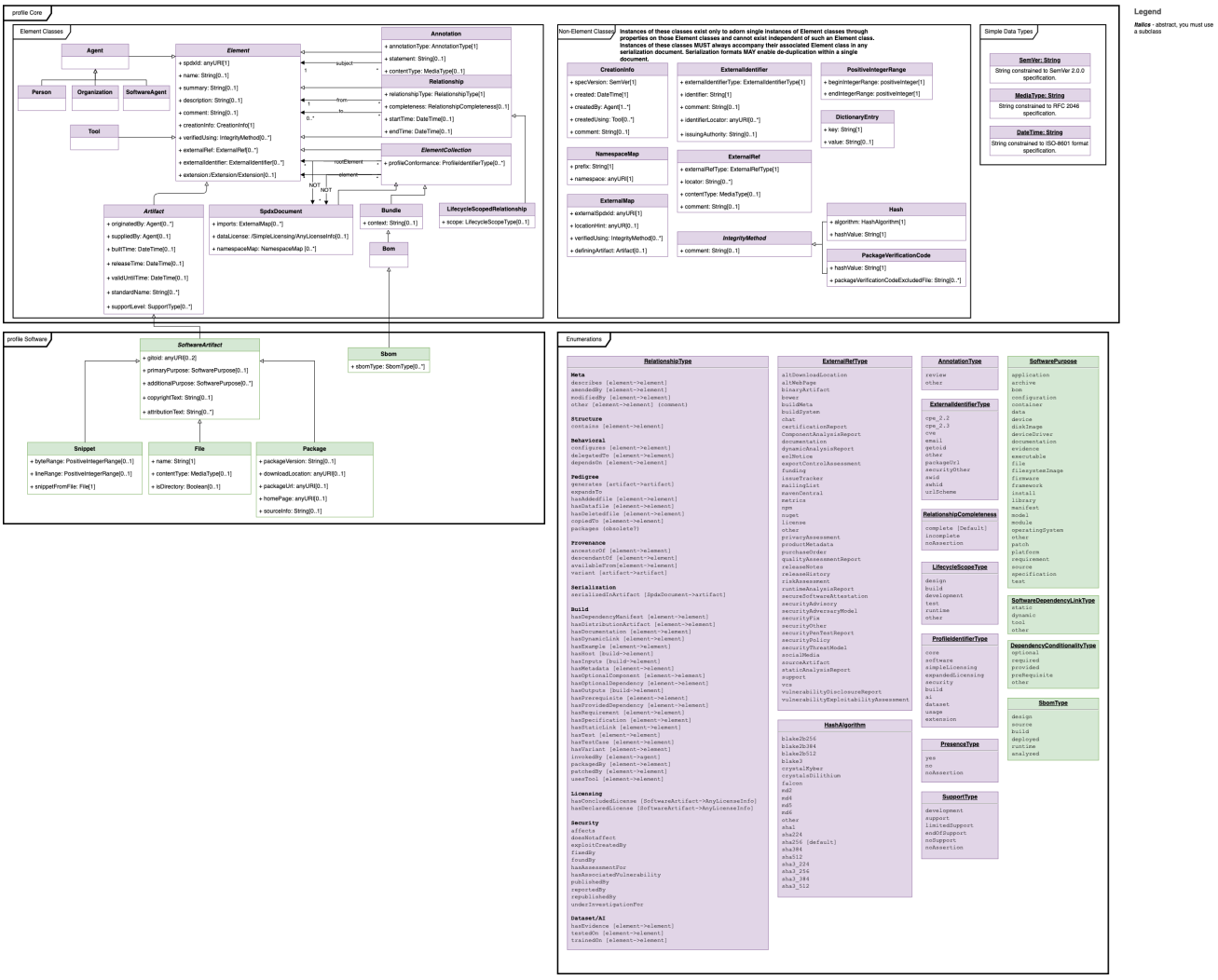
TODO: update for SPDXv3

C.1 Introduction

SPDX ® Vocabulary Specification

SPDX 3.0 Model SHACL

Version: 3.0



Annex D: SPDX license expressions (Normative)

D.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.

Annex E: Using SPDX license list short identifiers in source files (Informative)

TODO: update for SPDXv3

E.1 Introduction

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

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

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

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

E.2 Format for SPDX-License-Identifier

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

Annex F: Using SPDX to comply with Norms, Standards and Regulation (Informative)

F.1 Satisfying NTIA Minimum Elements for an SBOM using SPDX / US Executive Order 14028

US Executive Order 14028 in conjunction with the National Telecommunications and Information Administration (NTIA) outlined minimum elements for an SBOM. The minimum elements are detailed in NTIA's Framing Software Component Transparency: Establishing a Common Software Bill of Maternials and The Minimum Elements for a SBOM documents and summarized below:

SBOM Minimum Field	Description
Author Name	Author of the SBOM entry (this may not always be the supplier).
Supplier Name	Name or identity of the supplier of the component in the SBOM entry.
Component Name	Designation assigned to a unit of software defined by the original supplier.
Version String	Version used to identify a component.
Component Hash	A cryptographic hash to uniquely identify a component.
Unique Identifier	A unique identifier to help identify components or serve as a look-up key for relevant databases.
Relationship	Characterizing the relationship that an upstream component X is included in software Y.
Timestamp	Record of the date and time of the SBOM data assembly.

The SPDX Specification contains fields able to address each of the NTIA minimum required data fields.

NTIA SBOM Minimum Field	Satisfying SPDX field model location
Author Name	Core/Classes/CreationInfo.createdBy
Supplier Name	Core/Classes/Artifact.suppliedBy
Component Name	Software/Classes/Package.name inherited from Core/Classes/Element.name
Version String	Software/Classes/Package.packageVersion
Component Hash	Core/Classes/Element.verifiedUsing
Unique Identifier	Software/Classes/SoftwareArtifact.contentIdentifier for SPDX Software Artifacts or Software/Classes/Package.packageUrl if the packageUrl is considered to be unique, or Core/Classes/Element.externalIdentifier for resources outside the scope of SPDX-3.0 content

Annex G: Including Security Information in a SPDX document

The flexibility of SPDX 3.0 allows users to either link SBOMs to external security vulnerability data or to embed security vulnerability information in the SPDX 3.0 data format. For more details about the differences, read "Capturing Software Vulnerability Data in SPDX 3.0".

G.1 External References and External Identifiers

SPDX 3.0 has the concept of an **External Reference** for an Element which points to a general resource outside the scope of the SPDX-3.0 content that provides additional context or information about an Element.

The specification for External Reference types has many type options, a large handful of which pertain specifically to security use cases:

- cwe
- secureSoftwareAttestation
- securityAdvisory
- securityAdversaryModel
- securityFix
- securityOther
- securityPenTestReport
- securityPolicy
- securityThreatModel
- vulnerabilityDisclosureReport
- vulnerabilityExploitabilityAssessment

SPDX 3.0 also has the concept of **External Identifier** which should be used in cases where an identifier scheme exists and is already defined for an Element outside of SPDX-3.0.

There are several External Identifier types that may be used in a security context:

- cpe22
- cpe23
- cve

Annex H: SPDX Lite

H.1 Definition of the Lite profile

The Lite profile is designed to make it quick and easy to start a Software Bill of Materials in situations where a company may have limited capacity for introducing new items into its process.
The Lite profile captures the minimum set of information required for license compliance in the software supply chain. It contains information about the creation of the SBOM, package lists with licensing and other related items, and their relationships.

All elements in Lite profile are essential for complying with licenses. It is easy to use a SPDX document with the Lite profile for anyone who does not have enough knowledge about licensing information and easy to import license information from former versions of SPDX Lite format files.
The Lite profile offers the flexibility to be used either alone or in combination with other SPDX profiles as a SPDX document in the software supply chain.

H.2 Table of the Lite profile elements

A SPDX document with the Lite profile must include properties for each class listed in **Table H.1**. And `Cardinality 1..1` means a **REQUIRED** element, and the others **SHOULD** be filled in as much as possible if necessary.

Table H.1 — the Lite profile elements

1. For a /Core/SpdxDocument to be conformant with this profile, the following has to hold:

#	Property Name	Cardinality	Comments
1	/Core/SpdxDocument/spdxId	1..1	
2	/Core/SpdxDocument/name	0..1	
3	/Core/SpdxDocument/comment	0..1	
4	/Core/SpdxDocument/creationInfo	1..1	
5	/Core/SpdxDocument/verifiedUsing	0..1	This should be an object of /Core/Hash
6	/Core/SpdxDocument/element	1..*	MUST have at least one element
7	/Core/SpdxDocument/rootElement	1..1	This should be an object of /Core/Sbom
8	/Core/SpdxDocument/namespaceMap	0..*	

Creative Commons Attribution License 3.0 Unported

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1) **Definitions**

a. **"Adaptation"** means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. **"Collection"** means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

Community Specification License 1.0

The Purpose of this License. This License sets forth the terms under which 1) Contributor will participate in and contribute to the development of specifications, standards, best practices, guidelines, and other similar materials under this Working Group, and 2) how the materials developed under this License may be used. It is not intended for source code. Capitalized terms are defined in the License's last section.

1. Copyright.

1.1. Copyright License. Contributor grants everyone a non-sublicensable, perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as expressly stated in this License) copyright license, without any obligation for accounting, to reproduce, prepare derivative works of, publicly display, publicly perform, and distribute any materials it submits to the full extent of its copyright interest in those materials. Contributor also acknowledges that the Working Group may exercise copyright rights in the Specification, including the rights to submit the Specification to another standards organization.

1.2. Copyright Attribution. As a condition, anyone exercising this copyright license must include attribution to the Working Group in any derivative work based on materials developed by the Working Group. That attribution must include, at minimum, the material's name, version number, and source from where the materials were retrieved. Attribution is not required for implementations of the Specification.

2. Patents.

2.1. Patent License.

2.1.1. As a Result of Contributions.

2.1.1.1. As a Result of Contributions to Draft Specifications. Contributor grants Licensee a non-sublicensable, perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as expressly stated in this License) license to its Necessary Claims in 1) Contributor's Contributions and 2) to the Draft Specification that is within Scope as of the date of that Contribution, in both cases for Licensee's Implementation of the Draft Specification, except for those patent claims excluded by Contributor under Section 3.