INTERNATIONAL STANDARD

ISO/IEC 5962

First edition
2021-08

Information technology — SPDX® Specification V2.2.1

Technologies de l'information — Spécification SPDX® V2.2.1

Reference number
ISO/IEC 5962:2021(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by the Joint Development Foundation (JDF) (as SPDX® Specification V2.2.1) and drafted in accordance with its editorial rules. It was adopted, under the JTC 1 PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

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.

Information technology — SPDX® Specification V2.2.1

1 Scope

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

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

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

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

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

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

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

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

3 Terms and definitions

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

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/

3.1

annotations information section

section (3.9) type, an instance of which contains comments about an SPDX document, SPDX file, SPDX package, or SPDX snippet

3.2

field

a piece of information contained in a *section* (3.9)

3.3

file information section

section (3.9) type, an instance of which contains facts specific to files

3.4

other licensing information detected section

section (3.9) type, an instance of which contains a way to capture information about and refer to licenses that are not on the SPDX license List

3.5

package

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

3.6

package information section

section (3.9) type, an instance of which contains facts that are common properties of a *package*

3.7

relationships between SPDX elements information section

section (3.9) type, an instance of which contains information on how documents, *packages* (3.5), files and snippets relate to each other

3.8

review information section

section (3.9) type, an instance of which contains information about persons, organizations or tools that have reviewed a document

3.9

section

a part of this SPDX specification

3.10

snippet information section

section (3.9) type, an instance of which contains facts that are specific to a part of a file

3.11

SPDX document

collection of *section* (3.9) instances each of which contains information about software organized using the *SPDX format* (3.11)

3.12

SPDX document creation information section

section (3.9) type, an instance of which contains metadata that associates analysis results with a specific version of an *SPDX document* (3.11) and license for use, and provides information on how, when, and by whom the SPDX document was created

3.13

SPDX format

the data format defined by this document

3.14

sub-package

a package which is embedded in a larger package

4 Conformance

4.1 SPDX Current and Previous Versions

This edition has the version number 2.2.1 as part of its title. Although this is the first ISO edition of the SPDX Specification, earlier editions were published by the SPDX workgroup via the Linux Foundation. The SPDX Specification 2.2.1 was subsequently transposed into the Joint Development Foundation. [Those earlier editions are: 1.0 (August 2011), 1.1 (August 2012), 1.2 (October 2013), 2.0 (May 2015), 2.1 (November 2016), and 2.2 (May 2020).] Differences between this edition and earlier ones are reported in Annex I; see also [1].

4.2 Obsolete features

Over the life of a standard, some older approaches can become obsolete and are dropped from subsequent editions, possibly with a replacement approach being provided. Such action involves *deprecating* those outdated features. This edition identifies all currently deprecated features.

4.3 Alternate notation for some conformance requirements

This standard contains more than a few *cardinality assertions*, each of which indicates absolute, optional, or conditional requirements. Here are some examples:

— Cardinality: Mandatory, one.

— Cardinality: Optional, one or many.

- Cardinality: Mandatory, one if {condition} is true or {feature} omitted, zero (shall be omitted) if {condition} is false.
- Cardinality: 0..1
- Cardinality: 0..*
- Cardinality: 1..1
- Cardinality: 1..*

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

4.4 Standard data format requirements

The data format specification and recommendations are subject to the following constraints:

- Shall be in a human readable form.
- Shall be in a syntax that a software tool can read and write.
- Shall be suitable to be checked for syntactic correctness automatically, independent of how it was generated (human or tool).
- The SPDX document character set shall support UTF-8 encoding.
- Multiple serialization formats may be used to represent the information being exchanged. Current supported formats include:
    - **YAML 1.2** see: https://yaml.org/spec/1.2/spec.html
    - **JavaScript Object Notation** (JSON) see: ECMA-404
        - The JSON Schema for SPDX can be found in the SPDX Spec Git Repository Schema directory
    - **Resource Description Framework** (RDF also referred to as RDF/XML) see: https://www.w3.org/TR/rdf-syntax-grammar/
    - **tag:value** flat text file as described in this specification
    - **.xls** spreadsheets
- In addition to the supported formats, the following format is in development with a plan to complete the specification in the next release:
    - **Extensible Markup Language** (XML) see: https://www.w3.org/TR/2008/REC-xml-20081126/
- Interoperability between all the supported file formats shall be preserved. SPDX defines how to validate a document in each supported format, and how to translate a valid document without loss to each other supported format.

- Tags and format properties are case sensitive.
- Should be easy to recognize in a file system without opening the file. A suggested naming convention is:

Table 1 — Suggested naming convention

Format	Extension
tag:value	*.spdx
RDF	*.spdx.rdf
JSON	*.spdx.json
XML	*.spdx.xml
YAML	*.spdx.yaml or *.spdx.yml

- The convention in this specification is for the RDF examples to use `rdf:about="..."` to represent that a proper Universal Resource Indicator (URI) should be present.

4.5 Trademark Compliance

To be designated an SPDX document, a file shall comply with the requirements of the SPDX Trademark License (See the SPDX Trademark Page).

The official copyright notice that shall be used with any verbatim reproduction and/or distribution of this SPDX Specification 2.2.1 is:

"Official SPDX® Specification 2.2.1 Copyright © 2010-2020 Linux Foundation and its Contributors. Licensed under the Creative Commons Attribution License 3.0 Unported. All other rights are expressly reserved."

The official copyright notice that shall be used with any non-verbatim reproduction and/or distribution of this SPDX Specification 2.2.1, including without limitation any partial use or combining this SPDX Specification with another work, is:

"This is not an official SPDX Specification. Portions herein have been reproduced from SPDX® Specification 2.2.1 found at spdx.dev. These portions are Copyright © 2010-2020 Linux Foundation and its Contributors, and are licensed under the Creative Commons Attribution License 3.0 Unported by the Linux Foundation and its Contributors. All other rights are expressly reserved by Linux Foundation and its Contributors."

4.6 The SPDX Lite profile

Rather than conforming to this whole specification, an implementation may conform with SPDX Lite only, a profile that defines a subset of the SPDX specification. SPDX Lite aims at the balance between the SPDX standard and actual workflows in some industries. See Annex G for more information.

5 Composition of an SPDX document

5.1 What this specification covers

This document contains the specification for an SPDX document, which is made up of a set of one or more sections, instances of which contain information in the form of *fields*. The following subclauses introduce the different kinds of sections allowed. The fields for each kind of section are defined in the clause corresponding to that section. Within an SPDX document, sections may be organized, as follows:

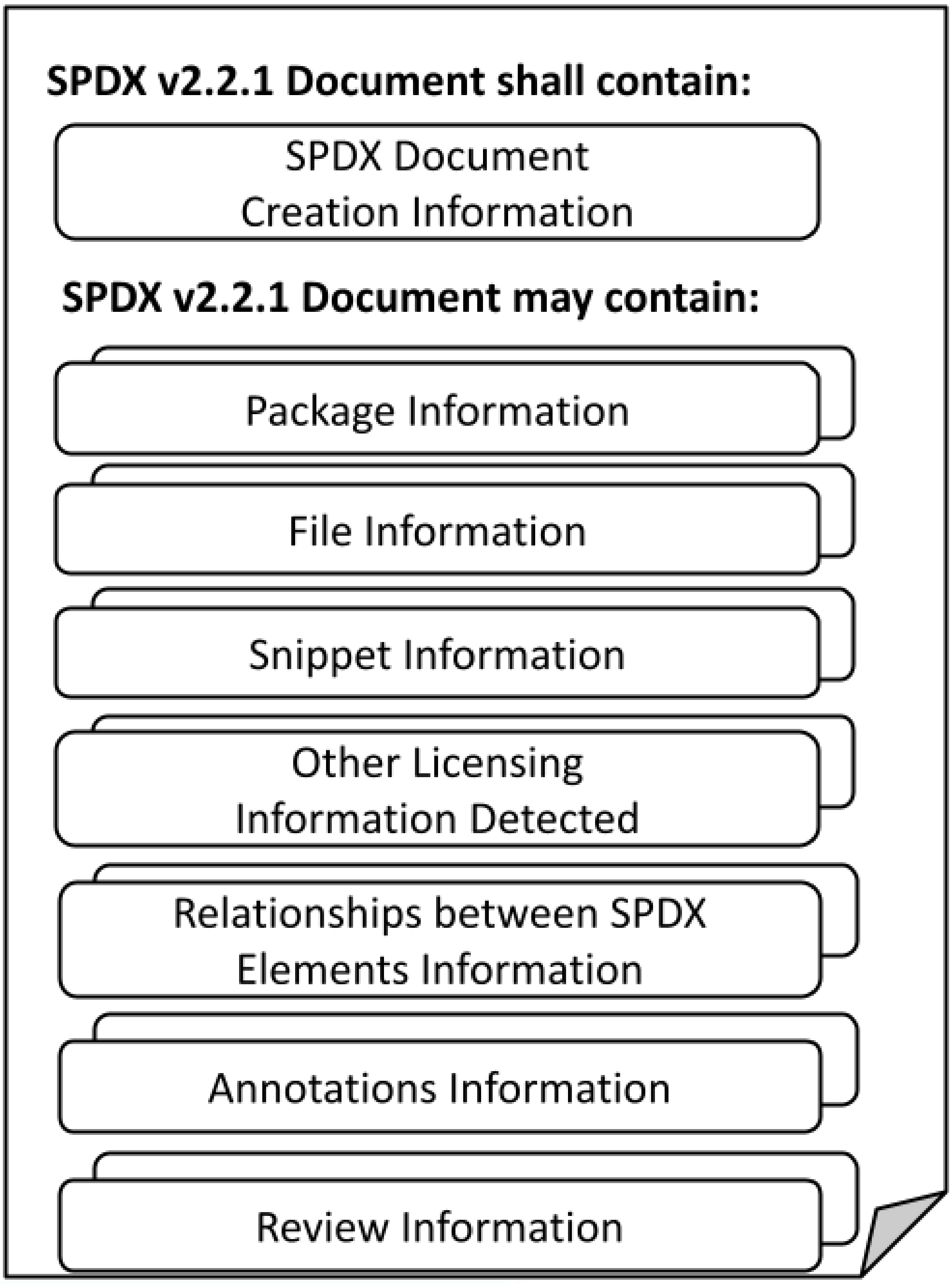
SPDX v2.2.1 Document shall contain:
SPDX Document Creation Information
SPDX v2.2.1 Document may contain:
Package Information
File Information
Snippet Information
Other Licensing Information Detected
Relationships between SPDX Elements Information
Annotations Information
Review Information

Figure 1 — SPDX document sections

The object model is illustrated by Annex C.

5.2 Sections

5.2.1 SPDX document creation information section

An instance of this section provides the necessary information for forward and backward compatibility for processing tools.

One instance shall be present for each SPDX document produced.

Cardinality: Mandatory, one.

See Clause 6 for details of the fields in this kind of section.

5.2.2 Package information section

If SPDX information is being used to describe packages, then one instance of the Package Information per package being described shall exist. It provides important meta information about the package as a whole. Packages are an abstract concept that can be used to refer to any distribution of software, typically consisting of one or more files and capable of containing sub-packages. Starting with SPDX 2.0, it is not necessary to have a package wrapping a set of files.

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. An SPDX document may, but is not required to, provide details about the individual files comprising a Package (see Clause 8).

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.

In an SPDX document, Relationship elements can be used to indicate relationships between Packages, such as dependency relationships.

Cardinality: Optional, one or many.

See Clause 7 for details of the fields in this kind of section.

In `tag:value` format, the order in which package and files occur is syntactically significant.

- A new Package Information section is denoted by the Package Name (7.1) field.
- All Package Information fields shall be grouped together before the start of a Files section (Clause 8), if file(s) are present.
- All files contained in a package shall immediately follow the applicable Package Information.

- A new Package Information section (via Package Name) denotes the start of another package.

- Sub-packages shall not be nested inside a Package Information section, but shall be separate and shall use a Relationship to clarify.

- Annotations and Relationships for the package may appear after the Package Information before any file information.

5.2.3 File information section

One instance of the File Information shall exist for each file in the software package. It provides important meta information about a given file including licenses and copyright. Starting with SPDX 2.0, it is not necessary to have a package wrapping a set of files.

Cardinality: Optional, one or many.

See Clause 8 for details of the fields in this kind of section.

When implementing `tag:value` format, the positioning of File elements is syntactically significant:

- Files are assumed to be associated with the Package Information that immediately precedes it, if a package exists.

- Presence of a new Package Information signals the end of the set of files associated with the preceding package, unless an explicit Relationship is used.

- If a package contains files, the File Information sections shall follow its Package Information section.

- If a File is not part of any package, it shall precede any Package Information section reference in the SPDX document.

- The first field to start off the description of a File shall be the File Name in `tag:value` format.

- File information is associated with the File Name that precedes it.

- Annotations on the file and Relationships from the file may appear after the file information, before the next file or Package Information section.

When implementing file information in RDF, the `spdx:hasFile` property is used to associate the package with the file.

5.2.4 Snippet information section

Snippets can optionally be used when a file is known to have some content that has been included from another original source. They are useful for denoting when part of a file may have been originally created under another license.

Each instance of Snippet Information shall be associated with a specific file in an SPDX document.

Cardinality: Optional, one or many.

See Clause 9 for details of the fields in this kind of section.

When implementing `tag:value` format, the positioning of Snippet elements is syntactically significant:

- If a File contains Snippets, the Snippet Information section shall follow a related File Information section (if it exists in the document).

- Presence of a new file or package section signals the end of the set of snippets associated with the original file, unless an explicit Relationship is used.
- The first field to start off the description of a Snippet shall be the Snippet Identifier in `tag:value` format.
- Annotations on the Snippet and Relationships from the Snippet may appear after the Snippet Information, before the next file or Package section.

5.2.5 Other licensing information detected section

This section is used for any detected, declared or concluded licenses that are NOT on the SPDX License List. For the most up-to-date version of the list, see https://spdx.org/licenses/. The SPDX License List can also be found in Annex A.

One instance shall be created for every unique license or licensing information being referenced that does not match one of the licenses on the SPDX License List.

Cardinality: Optional, one or many.

See Clause 10 for details of the fields in this kind of section.

5.2.6 Relationships between SPDX elements information section

Packages, files, and snippets are all considered to be SPDX Elements, and relationships can be made explicit between these SPDX elements by using the fields in this section.

Cardinality: Optional, one or many.

See Clause 11 for details of the fields in this kind of section.

5.2.7 Annotations information section

Annotations permit the addition of information to validate and clarify ambiguous SPDX Elements (Packages, Files or Snippets).

Cardinality: Optional, one or many.

See Clause 12 for details of the fields in this kind of section.

This section is now the preferred home for review information.

5.2.8 Review information section

The review information section is included for compatibility with SPDX 1.2, and is deprecated since SPDX 2.0. Any review information shall use an Annotation (as described in Clause 12) with an annotation type of `REVIEW`.

Review information may be added after the initial SPDX document has been created. The set of fields are optional and multiple instances may be added. Once a Reviewer entry is added, the Review Date associated with the review is mandatory. The Created date shall not be modified as a result of the addition of information regarding the conduct of a review. A Review Comments is optional.

See Clause 13 for details of the fields in this kind of section.

5.3 What this specification does not cover

This document does not address the following:

— Information that cannot be derived from an inspection (whether manual or using automated tools) of the package to be analyzed.

— How the data stored in an SPDX document is used by the recipient.

— Any identification of any patent(s) which may or may not relate to the package.

— Legal interpretation of the licenses or any compliance actions that have been or may need to be taken.

6 SPDX document creation information section

6.1 SPDX version field

6.1.1 Description

Provide a reference number that can be used to understand how to parse and interpret the rest of the file. It will enable both future changes to the specification and to support backward compatibility. The version number consists of a major and minor version indicator. The major field shall be incremented when incompatible changes between versions are made (one or more sections are created, modified or deleted). The minor field shall be incremented when backwards compatible changes are made. The metadata for the SPDX version field is shown in Table 2.

Table 2 — Metadata for the SPDX version field

Attribute	Value
Required	Yes
Cardinality	1..1
Format	`SPDX-M.N` where: • `M` is major version number • `N` is minor version number.

6.1.2 Intent

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.

6.1.3 Examples

EXAMPLE 1 Tag: `SPDXVersion:`

```
SPDXVersion: SPDX-2.2
```

EXAMPLE 2 RDF: Property `spdx:specVersion` in class `spdx:SpdxDocument`

```
<SpdxDocument rdf:about="...">
   <specVersion>SPDX-2.2</specVersion>
</SpdxDocument>
```

This specification uses the prefix `rdf:` to refer to the RDF/XML namespace:

```
http://www.w3.org/1999/02/22-rdf-syntax-ns#
```

6.2 Data license field

6.2.1 Description

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 document 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. The metadata for the data license field is shown in Table 3.

Table 3 — Metadata for the data license field

Attribute	Value
Required	Yes
Cardinality	1..1
Format	`CC0-1.0`

6.2.2 Intent

This is to alleviate any concern that content (the data or database) in an SPDX document is subject to any form of intellectual property right that could restrict the re-use of the information or the creation of another SPDX document for the same project(s). This approach avoids intellectual property and related restrictions over the SPDX document, however individuals can still contract with each other to restrict release of specific collections of SPDX documents (which map to software bill of materials) and the identification of the supplier of SPDX documents.

6.2.3 Examples

EXAMPLE 1 Tag: `DataLicense:`

```
DataLicense: CC0-1.0
```

EXAMPLE 2 RDF: Property `spdx:dataLicense` in class `spdx:SpdxDocument`

```
<SpdxDocument rdf:about="...">
  <dataLicense rdf:resource="http://spdx.org/licenses/CC0-1.0" />
</SpdxDocument>
```

6.3 SPDX identifier field

6.3.1 Description

Identify the current SPDX document which may be referenced in relationships by other files, packages internally and documents externally. To reference another SPDX document in total, this identifier should be used with the external document identifier preceding it. See Clause 11 for examples. The metadata for the SPDX identifier field is shown in Table 4.

Table 4 — Metadata for SPDX identifier field

Attribute	Value
Required	Yes
Cardinality	1..1
Format	`SPDXRef-DOCUMENT`

6.3.2 Intent

Provide a way for the document to refer to itself in relationship to other elements.

6.3.3 Examples

EXAMPLE 1 Tag: `SPDXID:`

```
SPDXID: SPDXRef-DOCUMENT
```

EXAMPLE 2 RDF:

The URI for the document is the document namespace appended by

```
#SPDXRef-DOCUMENT
```

```
<spdx:SpdxDocument
  rdf:about="http://spdx.org/spdxdocs/spdx-example-444504E0-4F89-41D3-9A0
C-0305E82C33123#SPDXRef-DOCUMENT">
   ...
</spdx:SpdxDocument>
```

6.4 Document name field

6.4.1 Description

Identify name of this document as designated by creator. The metadata for the document name field is shown in Table 5.

Table 5 — Metadata for the document name field

Attribute	Value
Required	Yes
Cardinality	1..1
Format	Single line of text.

6.4.2 Intent

Here, the name of each document is an important convention and easier to refer to than the URI.

6.4.3 Examples

EXAMPLE 1 Tag: `DocumentName:`

```
DocumentName: glibc-v2.3

DocumentName: ubuntu-14.04
```

EXAMPLE 2 RDF: Property `spdx:name` in class `spdx:SpdxDocument`

```
<SpdxDocument rdf:about="...">
  <name>glibc-v2.3</name>
</SpdxDocument>

<SpdxDocument rdf:about="...">
  <name>ubuntu-14.04</name>
</SpdxDocument>
```

6.5 SPDX document namespace field

6.5.1 Description

Provide an SPDX document-specific namespace as a unique absolute Uniform Resource Identifier (URI) as specified in RFC-3986, with the exception of the '#' delimiter. The SPDX document URI shall not contain a URI "part" (e.g. the "#" character), since the '#' is used in SPDX element URIs (packages, files, snippets, etc) to separate the document namespace from the element's SPDX identifier. Additionally, a scheme (e.g. "https:") is required.

The URI shall be unique for the SPDX document including the specific version of the SPDX document. If the SPDX document is updated, thereby creating a new version, a new URI for the updated document shall be used. There may only be one URI for an SPDX document and only one SPDX document for a given URI. The metadata for the SPDX document namespace field is shown in Table 6.

Table 6 — Metadata for the SPDX document namespace field

Attribute	Value
Required	Yes
Cardinality	1..1
Format	Unique absolute Uniform Resource Identifier (URI) as specified in RFC-3986, with the following exceptions: The SPDX document URI cannot contain a URI "part" (e.g. the # delimiter), since the # is used to uniquely identify SPDX element identifiers. The URI shall contain a scheme (e.g., `https:`). The URI shall be unique for the SPDX document including the specific version of the SPDX document. If the SPDX document is updated, thereby creating a new version, a new URI for the updated document shall be used. There can only be one URI for an SPDX document and only one SPDX document for a given URI.

6.5.2 Intent

The URI provides an unambiguous mechanism for other SPDX documents to reference SPDX elements within this SPDX document. See 6.6 for a description on how external documents are referenced. Although it is not required, the URI can be constructed in a way which provides information on how the SPDX document can be found. For example, the URI can be a URL referencing the SPDX document itself, if it is available on the internet. A best practice for creating the URI for SPDX documents available on the public internet is `http://[CreatorWebsite]/[pathToSpdx]/[DocumentName]-[UUID]` where:

- `CreatorWebsite` is a website hosted by the creator of the document. (e.g. an SPDX document provided by SPDX would be spdx.org)
- `PathToSpdx` is a path to where SPDX documents are stored on the website (e.g. /spdx/spdxdocs)
- `DocumentName` is a name given to the SPDX document itself, typically the (set of) package name(s) followed by the version. (See 6.4.)
- `UUID` is a universally unique identifier. The UUID could be a version 4 random UUID which can be generated from the Online UUID Generator or a version 5 UUID generated from a sha1 checksum known to be unique for this specific SPDX document version.
- If the creator does not own their own website, a default SPDX CreatorWebsite and PathToSpdx can be used `spdx.org/spdxdocs`. Note that the SPDX documents are not currently stored or accessible on this website. The URI is only used to create a unique ID following the above conventions.

NOTE The URI does not have to be accessible. It is only intended to provide a unique ID. In many cases, the URI will point to a web accessible document, but this should not be assumed to be the case.

6.5.3 Examples

EXAMPLE 1 Tag: `DocumentNamespace:`

```
DocumentNamespace: http://spdx.org/spdxdocs/spdx-tools-v1.2-3F2504E0-4F89
-41D3-9A0C-0305E82...
```

EXAMPLE 2 RDF: The unique ID is the URI for the SPDX document

```
<SpdxDocument rdf:about="http://spdx.org/spdxdocs/spdx-tools-v1.2-3F2504E
0-4F89-41D3-9A0C-0305E82...">
    <rdfs:comment>This document was created using SPDX 2.0 using
    licenses from the web site.</rdfs:comment>
</SpdxDocument>
```

This specification uses the prefix `rdfs:` to refer to the RDF Schema namespace:

```
http://www.w3.org/2000/01/rdf-schema#
```

6.6 External document references field

6.6.1 Description

Identify any external SPDX documents referenced within this SPDX document. The metadata for the external document references field is shown in Table 7.

Table 7 — Metadata for the external document references field

Attribute	Value
Required	No
Cardinality	1..*
Format	DocumentRef-`[idstring] [SPDX document URI] [Checksum]` where `[idstring]` is a unique string containing letters, numbers, `.`, – and/or +. `[SPDX document URI]` is the unique ID for the external document as defined in 6.5 of that referenced document, `[Checksum]` is a checksum of the external document following the checksum format defined in 8.4.

6.6.2 Intent

SPDX elements within this document may be related to other SPDX elements referenced from external SPDX documents. An SPDX element could be a snippet, file, package, license reference or SPDX document.

6.6.3 Examples

EXAMPLE 1 Tag: `ExternalDocumentRef:`

```
ExternalDocumentRef:DocumentRef-spdx-tool-1.2 http://spdx.org/spdxdocs/sp
dx-tools-v1.2-3F2504E0-4F89-41D3-9A0C-0305E82C3301 SHA1: d6a770ba38583ed4
bb4525bd96e50461655d2759
```

EXAMPLE 2 RDF: Property `spdx:externalDocumentRef` in class `spdx:SpdxDocument range ExternalDocumentRef`.

The ExternalDocumentRef contains two properties:

- spdxDocument - the SpdxDocument being referenced
- checksum - the checksum of the referenced SPDX document

```
<SpdxDocument rdf:about="...">
  <externalDocumentRef rdf:ID="DocumentRef-spdx-tool-1.2">
    <ExternalDocumentRef>
        <spdxDocument rdf:about="http://spdx.org/spdxdocs/spdx-tools-v1.2
-3F2504E0-4F89-41D3-9A0C-0305E82..." />
        <checksum>
            <Checksum>
                <algorithm rdf:resource="checksumAlgorithm_sha1"/>
                <checksumValue>d6a770ba38583ed4bb4525bd96e50461655d2758
                </checksumValue>
            </Checksum>
        </checksum>
    </ExternalDocumentRef>
  </externalDocumentRef>
</SpdxDocument>
```

NOTE In RDF, a namespace can be created for the external document reference if a short form name for the external reference is desired.

6.7 License list version field

6.7.1 Description

An optional field for creators of the SPDX document to provide the version of the SPDX License List used when the SPDX document was created. The metadata for the license list version field is shown in Table 8.

Table 8 — Metadata for the license list version field

Attribute	Value
Required	No
Cardinality	1..1
Format	`M.N` where: `M` is major version number `N` is minor version number.

6.7.2 Intent

Recognizing that licenses are added to the SPDX License List with each subsequent version, the intent is to provide recipients of the SPDX document with the version of the SPDX License List used. This anticipates that in the future, an SPDX document might have used a version of the SPDX License List that is older than the then current one.

6.7.3 Examples

EXAMPLE 1 Tag: `LicenseListVersion:`

```
LicenseListVersion: 3.8
```

EXAMPLE 2 RDF: Property `licenseListVersion` in class `spdx:CreationInfo`

```
<CreationInfo>
    <licenseListVersion>3.8</licenseListVersion>
</CreationInfo>
```

6.8 Creator field

6.8.1 Description

Identify who (or what, in the case of a tool) created the SPDX document. If the SPDX document was created by an individual, indicate the person's name. If the SPDX document was created on behalf of a company or organization, indicate the entity name. If the SPDX document was created using a software tool, indicate the name and version for that tool. If multiple participants or tools were involved, use multiple instances of this field. Person name or organization name may be designated as “anonymous” if appropriate. The metadata for the creator field is shown in Table 9.

Table 9 — Metadata for the creator field

Attribute	Value
Required	Yes
Cardinality	1..*

Format	Single line of text with the following keywords:
	`"Person: person name" and optional "(email)"`
	`"Organization: organization" and optional "(email)"`
	`"Tool: toolidentifier-version"`

6.8.2 Intent

Here, the generation method will assist the recipient of the SPDX document in assessing the general reliability/accuracy of the analysis information.

6.8.3 Examples

EXAMPLE 1 Tag: `Creator:`

```
Creator: Person: Jane Doe ()
Creator: Organization: ExampleCodeInspect ()
Creator: Tool: LicenseFind-1.0
```

EXAMPLE 2 RDF: Property `spdx:creator` in class `spdx:CreationInfo`

```
<CreationInfo>
    <creator> Person: Jane Doe () </creator>
    <creator> Organization: ExampleCodeInspect () </creator>
    <creator> Tool: LicenseFind-1.0 </creator>
</CreationInfo>
```

6.9 Created field

6.9.1 Description

Identify when the SPDX document was originally created. The date is to be specified according to combined date and time in UTC format as specified in ISO 8601 standard. This field is distinct from the fields in Clause 12, which involves the addition of information during a subsequent review. The metadata for the created field is shown in Table 10.

Table 10 — Metadata for the created field

Attribute	Value
Required	Yes
Cardinality	1..1
Format	`YYYY-MM-DDThh:mm:ssZ` where: • `YYYY` is year • `MM` is month with leading zero • `DD` is day with leading zero • `T` is delimiter for time • `hh` is hours with leading zero in 24-hour time • `mm` is minutes with leading zero • `ss` is seconds with leading zero • `Z` is universal time indicator

6.9.2 Intent

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

6.9.3 Examples

EXAMPLE 1 Tag: `Created:`

```
Created: 2010-01-29T18:30:22Z
```

EXAMPLE 2 RDF: Property `spdx:created` in class `spdx:CreationInfo`

```
<CreationInfo>
    <created> 2010-01-29T18:30:22Z </created>
</CreationInfo>
```

6.10 Creator comment field

6.10.1 Description

An optional field for creators of the SPDX document to provide general comments about the creation of the SPDX document or any other relevant comment not included in the other fields. The metadata for the Creator comment field is shown in Table 11.

Table 11 — Metadata for the Creator comment field

Attribute	Value
Required	No
Cardinality	1..1
Format	Free form text that can span multiple lines. In `tag:value` format this is delimited by `<text> .. </text>`, in RDF, it is delimited by `<rdfs:comment>`.

6.10.2 Intent

Here, the intent is to provide recipients of the SPDX document with comments by the creator of the SPDX document.

6.10.3 Examples

EXAMPLE 1 Tag: `CreatorComment:`

```
CreatorComment: <text>This SPDX document was created by a combination of
using a free tool, as indicated above, and manual analysis by several aut
hors of the code.</text>
```

EXAMPLE 2 RDF: Property `rdfs:comment` in class `spdx:CreationInfo`

```
<CreationInfo>
    <rdfs:comment>This SPDX document was created by a combination of
    using a free tool, as indicated above, and manual analysis
    by several authors of the code.</rdfs:comment>
</CreationInfo>
```

6.11 Document comment field

6.11.1 Description

An optional field for creators of the SPDX document content to provide comments to the consumers of the SPDX document. The metadata for the document comment field is shown in Table 12.

Table 12 — Metadata for the document comment field

Attribute	Value
Required	No
Cardinality	1..1
Format	Free form text that can span multiple lines. In `tag:value` format this is delimited by `<text> .. </text>`, in RDF, it is delimited by `<rdfs:comment>`.

6.11.2 Intent

Here, the intent is to provide readers/reviewers with comments by the creator of the SPDX document about the SPDX document.

6.11.3 Examples

EXAMPLE 1 Tag: `DocumentComment:`

```
DocumentComment: <text>This document was created using SPDX 2.0, version
2.3 of the SPDX License List and refering to licenses in file MyCompany.A
pproved.Licenses.spdx.</text>
```

EXAMPLE 2 RDF: Property `rdfs:comment` in class `SpdxDocument`

```
<SpdxDocument rdf:about="...">
    <rdfs:comment>
      This document was created using SPDX 2.0, version 2.3 of the SPDX
      License List and refering to licenses in file
      MyCompany.Approved.Licenses.spdx.
    </rdfs:comment>
</SpdxDocument>
```

7 Package information section

7.1 Package name field

7.1.1 Description

Identify the full name of the package as given by the Package Originator (7.6). The metadata for the SPDX version field is shown in Table 13.

Table 13 — Metadata for the package name field

Attribute	Value
Required	Yes
Cardinality	1..1
Format	Single line of text.

7.1.2 Intent

The name of each package is an important conventional technical identifier to be maintained for each package.

7.1.3 Examples

EXAMPLE 1 Tag: `PackageName:`

```
PackageName: glibc
```

EXAMPLE 2 RDF: Property `spdx:name` in class `spdx:Package`

```
<Package rdf:about="...">
    <name>glibc</name>
</Package>
```

7.2 Package SPDX identifier field

7.2.1 Description

Uniquely identify any element in an SPDX document which may be referenced by other elements. These may be referenced internally and externally with the addition of the SPDX document identifier. The metadata for the SPDX identifier field is shown in Table 14.

Table 14 — Metadata for the SPDX identifier field

Attribute	Value
Required	Yes
Cardinality	1..1
Format	"SPDXRef-"`[idstring]` where `[idstring]` is a unique string containing letters, numbers, `.`, and/or `-`.

7.2.2 Intent

There may be several versions of the same package within an SPDX document. Each element needs to be able to be referred to uniquely so that relationships between elements can be clearly articulated.

7.2.3 Examples

EXAMPLE 1 Tag: `SPDXID:`

```
SPDXID: SPDXRef-1
```

EXAMPLE 2 RDF: The URI for the element will follow the form:

```
[SPDX document namespace]#[SPDX identifier]
```

See 6.5 for the definition of the SPDX document namespace and 6.3 for the definition of the SPDX identifier

Using `xml:base`:

```
<rdf:RDF xml:base="http://acme.com/spdxdocs/acmeproj/v1.2/1BE2A4FF-5F1A-4
8D3-8483-28A9B0349A1B">
    ...
    <Package rdf:about="#SPDXRef-1">
    ...
    </Package>
</rdf:RDF>
```

Using document URI:

```
<Package rdf:about="http://acme.com/spdxdocs/acmeproj/v1.2/1BE2A4FF-5F1A-
48D3-8483-28A9B0349A1B#SPDXRef-1">
    ...
</Package>
```

7.3 Package version field

7.3.1 Description

Identify the version of the package. The metadata for the package version field is shown in Table 15.

Table 15 — Metadata for the package version field

Attribute	Value
Required	No
Cardinality	1..1
Format	Single line of text.

7.3.2 Intent

The versioning of a package is a useful for identification purposes and for indicating later changes of the package version.

7.3.3 Examples

EXAMPLE 1 Tag: `PackageVersion:`

```
PackageVersion: 2.11.1
```

EXAMPLE 2 RDF: Property `spdx:versionInfo` in class `spdx:Package`

```
<Package rdf:about="...">
    ...
    <versionInfo>2.11.1</versionInfo>
    ...
</Package>
```

7.4 Package file name field

7.4.1 Description

Provide the actual file name of the package, or path of the directory being treated as a package. This may include the packaging and compression methods used as part of the file name, if appropriate. The metadata for the package file name field is shown in Table 16.

Table 16 — Metadata for the package file name field

Attribute	Value
Required	No
Cardinality	1..1
Format	Single line of text.

7.4.2 Intent

The actual file name of the compressed file containing the package may be a significant technical element that needs to be included with each package identification information. If a grouping, like a set of files in a sub-directory, is being treated as a package, the sub-directory name may be appropriate to provide. Sub-directory name is preceded with a ./. See RFC 3986 for syntax.

7.4.3 Examples

EXAMPLE 1 Tag: `PackageFileName:`

```
PackageFileName: glibc-2.11.1.tar.gz
```

Sub-directory being treated as a package:

```
PackageFileName: ./myrootdir/mysubdir1
```

EXAMPLE 2 RDF: Property `spdx:packageFileName` in class `spdx:Package`

```
<Package rdf:about="...">
    ...
    <packageFileName>glibc 2.11.1.tar.gz</packageFileName>
    ...
</Package>
```

Sub-directory being treated as a package:

```
<Package rdf:about="...">
   ...
   <packageFileName>./myrootdir/mysubdir1</packageFileName>
   ...
</Package>
```

7.5 Package supplier field

7.5.1 Description

Identify the actual distribution source for the package/directory identified in the SPDX document. This might or might not be different from the originating distribution source for the package. The name of the Package Supplier shall be an organization or recognized author and not a web site. For example, SourceForge is a host website, not a supplier, the supplier for https://sourceforge.net/projects/bridge/ is "The Linux Foundation."

Use NOASSERTION if:

— the SPDX document creator has attempted to but cannot reach a reasonable objective determination;

— the SPDX document creator has made no attempt to determine this field; or

— the SPDX document creator has intentionally provided no information (no meaning should be implied by doing so).

The metadata for the package supplier field is shown in Table 17.

Table 17 — Metadata for the package supplier field

Attribute	Value
Required	No
Cardinality	1..1
Format	Single line of text with one of the following keywords | NOASSERTION • `Person`: person name and optional `(<email>)` • `Organization`: organization name and optional `(<email>)`

7.5.2 Intent

Assist with understanding the point of distribution for the code in the package. This field is vital for ensuring that downstream package recipients can address any ambiguity or concerns that might arise with the information in the SPDX document or the contents of the package it documents.

7.5.3 Examples

EXAMPLE 1 Tag: `PackageSupplier:`

```
PackageSupplier: Person: Jane Doe (jane.doe@example.com)
```

EXAMPLE 2 RDF: Property `spdx:supplier` in class `spdx:Package`

```
<Package rdf:about="...">
    ...
    <supplier>Person: Jane Doe (jane.doe@example.com)</supplier>
    ...
</Package>
```

7.6 Package originator field

7.6.1 Description

If the package identified in the SPDX document originated from a different person or organization than identified as Package Supplier (see 7.5 above), this field identifies from where or whom the package originally came. In some cases, a package may be created and originally distributed by a different third party than the Package Supplier of the package. For example, the SPDX document identifies the package glibc and Red Hat as the Package Supplier, but the Free Software Foundation is the Package Originator.

Use `NOASSERTION` if:

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

The metadata for the package originator field is shown in Table 18.

Table 18 — Metadata for the package originator field

Attribute	Value
Required	No
Cardinality	1..1
Format	Single line of text with one of the following keywords | `NOASSERTION` • `Person:` person name and optional `(<email>)` • `Organization:` organization name and optional `(<email>)`

7.6.2 Intent

Assist with understanding the point of origin of the code in the package. This field is vital for understanding who originally distributed a package and should help in addressing any ambiguity or concerns that might arise with the information in the SPDX document or the contents of the Package it documents.

7.6.3 Examples

EXAMPLE 1 Tag: `PackageOriginator:`

```
PackageOriginator: Organization: ExampleCodeInspect (contact@example.com)
```

EXAMPLE 2 RDF: Property `spdx:originator` in class `spdx:Package`

```
<Package rdf:about="...">
    <originator>Organization: ExampleCodeInspect
    (contact@example.com)</originator>
</Package>
```

7.7 Package download location field

7.7.1 Description

This section identifies the download Universal Resource Locator (URL), or a specific location within a version control system (VCS) for the package at the time that the SPDX document was created.

Use:

— `NONE` if there is no download location whatsoever.

— `NOASSERTION` if:

  — the SPDX document creator has attempted to but cannot reach a reasonable objective determination;

  — the SPDX document creator has made no attempt to determine this field; or

  — the SPDX document creator has intentionally provided no information (no meaning should be implied by doing so).

The metadata for the package download location field is shown in Table 19.

Table 19 — Metadata for the package download location field

Attribute	Value
Required	Yes
Cardinality	1..1
Format	Uniform resource locator | VCS location | `NONE` | `NOASSERTION` For version-controlled files, the VCS location syntax is similar to a URL and has the: `<vcs_tool>+<transport>://<host_name>[/<path_to_repository>][@<revision_tag_or_branch>][#<sub_path>]` This VCS location compact notation (inspired and mostly adopted from pip as of 2015-02-20) supports referencing locations in version control systems such as Git, Mercurial, Subversion and Bazaar, and specifies the type of VCS tool using url prefixes: `git+`, `hg+`, `bzr+`, `svn+` and specific transport schemes such as SSH or HTTPS. Specifying sub-paths, branch names, a commit hash, a revision or a tag name is recommended, and supported using the `@` delimiter for commits and the `#` delimiter for sub-paths. Using user names and password in the `<host_name>` is not supported and should be considered as an error. User access control to URLs or VCS repositories shall be handled outside of an SPDX document. In VCS location compact notations, the trailing slashes in `<host_name>`, `<path_to_repository>` are not significant. Leading and trailing slashes in `<sub_path>` are not significant.

7.7.2 Intent

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

7.7.3 Examples

EXAMPLE 1 Tag: `PackageDownloadLocation:`

If ambiguous:

```
PackageDownloadLocation: NOASSERTION

PackageDownloadLocation: NONE
```

For a plain URL:

```
PackageDownloadLocation: http://ftp.gnu.org/gnu/glibc/glibc-ports-2.15.ta
r.gz
```

For Git:

SPDX supported schemes are: `git`, `git+git`, `git+https`, `git+http`, and `git+ssh`. `git` and `git+git` are equivalent.

Here are the supported forms:

```
PackageDownloadLocation: git://git.myproject.org/MyProject

PackageDownloadLocation: git+https://git.myproject.org/MyProject.git

PackageDownloadLocation: git+http://git.myproject.org/MyProject

PackageDownloadLocation: git+ssh://git.myproject.org/MyProject.git

PackageDownloadLocation: git+git://git.myproject.org/MyProject

PackageDownloadLocation: git+git@git.myproject.org:MyProject
```

To specify a sub-path to a file or directory inside a repository use the `#` delimiter:

```
PackageDownloadLocation: git://git.myproject.org/MyProject#src/somefile.c

PackageDownloadLocation: git+https://git.myproject.org/MyProject#src/Clas
s.java
```

To specify branch names, a commit hash or a tag name, use the `@` delimiter:

```
PackageDownloadLocation: git://git.myproject.org/MyProject.git@master

PackageDownloadLocation: git+https://git.myproject.org/MyProject.git@v1.0

PackageDownloadLocation: git://git.myproject.org/MyProject.git@da39a3ee5e
6b4b0d3255bfef95601890afd80709
```

Sub-paths and branch names or commit hash can be combined too:

```
PackageDownloadLocation: git+https://git.myproject.org/MyProject.git@mast
er#/src/MyClass.cpp

PackageDownloadLocation: git+https://git.myproject.org/MyProject@da39a3ee
5e6b4b0d3255bfef95601890afd80709#lib/variable.rb
```

For Mercurial:

SPDX supported schemes are: `hg+http`, `hg+https`, `hg+static-http`, and `hg+ssh`.

The supported forms are:

```
PackageDownloadLocation: hg+http://hg.myproject.org/MyProject

PackageDownloadLocation: hg+https://hg.myproject.org/MyProject

PackageDownloadLocation: hg+ssh://hg.myproject.org/MyProject
```

To specify a sub-path to a file or directory inside a repository use the `#` delimiter:

```
PackageDownloadLocation: hg+https://hg.myproject.org/MyProject#src/somefile.c
```

```
PackageDownloadLocation: hg+https://hg.myproject.org/MyProject#src/Class.java
```

To pass branch names, a commit hash, a tag name or a local branch name use the `@` delimiter:

```
PackageDownloadLocation: hg+https://hg.myproject.org/MyProject@da39a3ee5e6b
```

```
PackageDownloadLocation: hg+https://hg.myproject.org/MyProject@2019
```

```
PackageDownloadLocation: hg+https://hg.myproject.org/MyProject@v1.0
```

```
PackageDownloadLocation: hg+https://hg.myproject.org/MyProject@special_feature
```

Sub-paths and branch names or commit hash can be combined too:

```
PackageDownloadLocation: hg+https://hg.myproject.org/MyProject@master#/src/MyClass.cpp
```

```
PackageDownloadLocation: hg+https://hg.myproject.org/MyProject@da39a3ee5e6b#lib/variable.rb
```

For Subversion:

SPDX supported schemes are: `svn`, `svn+svn`, `svn+http`, `svn+https`, `svn+ssh`. `svn` and `svn+svn` are equivalent.

The supported forms are:

```
PackageDownloadLocation: svn://svn.myproject.org/svn/MyProject
```

```
PackageDownloadLocation: svn+svn://svn.myproject.org/svn/MyProject
```

```
PackageDownloadLocation: svn+http://svn.myproject.org/svn/MyProject/trunk
```

```
PackageDownloadLocation: svn+https://svn.myproject.org/svn/MyProject/trunk
```

To specify a sub-path to a file or directory inside a repository use the `#` delimiter:

```
PackageDownloadLocation: svn+https://svn.myproject.org/MyProject#src/somefile.c
```

```
PackageDownloadLocation: svn+https://svn.myproject.org/MyProject#src/Class.java
```

This support is less important for SVN since the URL path can also contain sub-paths; this two forms are equivalent:

```
PackageDownloadLocation: svn+https://svn.myproject.org/MyProject/trunk#sr
c/somefile.c

PackageDownloadLocation: svn+https://svn.myproject.org/MyProject/trunk/sr
c/somefile.c
```

You can specify a revision using the `@` delimiter:

```
PackageDownloadLocation: svn+https://svn.myproject.org/svn/MyProject/trun
k@2019
```

Sub-paths and revisions can be combined too:

```
PackageDownloadLocation: svn+https://svn.myproject.org/MyProject@123#/src
/MyClass.cpp

PackageDownloadLocation: svn+https://svn.myproject.org/MyProject/trunk@12
34#lib/variable/variable.rb
```

For Bazaar:

SPDX supported schemes are: `bzr+http`, `bzr+https`, `bzr+ssh`, `bzr+sftp`, `bzr+ftp`, and `bzr+lp`.

The supported forms are:

```
PackageDownloadLocation: bzr+https://bzr.myproject.org/MyProject/trunk

PackageDownloadLocation: bzr+http://bzr.myproject.org/MyProject/trunk

PackageDownloadLocation: bzr+sftp://myproject.org/MyProject/trunk

PackageDownloadLocation: bzr+ssh://myproject.org/MyProject/trunk

PackageDownloadLocation: bzr+ftp://myproject.org/MyProject/trunk

PackageDownloadLocation: bzr+lp:MyProject
```

To specify a sub-path to a file or directory inside a repository use the `#` delimiter:

```
PackageDownloadLocation: bzr+https://bzr.myproject.org/MyProject/trunk#sr
c/somefile.c

PackageDownloadLocation: bzr+https://bzr.myproject.org/MyProject/trunk#sr
c/Class.java
```

You can specify a revision or tag using the `@` delimiter:

```
PackageDownloadLocation: bzr+https://bzr.myproject.org/MyProject/trunk@20
19

PackageDownloadLocation: bzr+http://bzr.myproject.org/MyProject/trunk@v1.
0
```

Sub-paths and revisions can be combined too:

```
PackageDownloadLocation: bzr+https://bzr.myproject.org/MyProject/trunk@20
19#src/somefile.c
```

EXAMPLE 2 RDF: Property `spdx:downloadLocation` in class `spdx:Package`

```
<Package rdf:about="...">
    <downloadLocation>http://ftp.gnu.org/gnu/glibc/glibc-ports-2.15.tar.g
z</downloadLocation>
</Package>

<Package rdf:about="...">
    <downloadLocation>
        git+https://git.myproject.org/MyProject.git@v10.0#src/lib.c
    </downloadLocation>
</Package>

<Package rdf:about="...">
    <downloadLocation rdf:resource="spdx:noassertion"/>
</Package>

<Package rdf:about="...">
    <downloadLocation rdf:resource="spdx:none"/>
</Package>
```

7.8 Files analyzed field

7.8.1 Description

Indicates whether the file content of this package has been available for or subjected to analysis when creating the SPDX document. If `false`, indicates packages that represent metadata or URI references to a project, product, artifact, distribution or a component. If `false`, the package shall not contain any files. The metadata for the files analyzed field is shown in Table 20.

Table 20 — Metadata for the files analyzed field

Attribute	Value
Required	No. If omitted, the default value of `true` is assumed.
Cardinality	1..1
Format	Boolean

7.8.2 Intent

A package can refer to a project, product, artifact, distribution or a component that is external to the SPDX document.

Some examples:

a) **A bundle of external products**: Package A can be metadata about Packages and their dependencies. It may also be a loosely organized manifest of references to Packages involved in a product or project. Build or execution may transitively discover more Packages and dependencies. All of these referenced Packages can have their own SPDX documents. In this case, Package A may be defined with its File Analyzed attribute set to `false`. Package A includes External Document References to SPDX documents containing Packages referenced in all the available relationships. The Relationships section then relates the SPDX documents and contained SPDX elements with appropriate semantics per the dependencies in the scope of Package A.

b) **Package relation to external product**: Package A can have a STATIC_LINK relationship to Package B, but the binary representation of Package B is furnished by the build process and thus not contained in the file list of Package A. In this case, Package B needs to be defined with its Files Analyzed attribute set to false and all the other attributes subject to the subsequently defined constraints. Then, the relationship between Package A and Package B can be documented as described in Clause 11.

c) **File derived from external product**: Package A contains multiple files derived from an outside project. Rather than use the `artifactOf*` attributes (F.9-4.11) to describe the relation of these files to their project, the outside project can be represented by another package, Package B, whose `FilesAnalyzed` (7.8) attribute is set to `false`. Each of the binary files can then have a relationship to package B (Clause 10). This allows the outside project to be represented by a single SPDX identifier (the identifier of Package B). It also allows the relationship(s) between the outside project and each of the files be represented in much more detail.

7.8.3 Examples

EXAMPLE 1 Tag: `FilesAnalyzed`

```
FilesAnalyzed: false
```

EXAMPLE 2 RDF: Property `spdx:filesAnalyzed` in class `spdx:Package`

```
<Package rdf:about="...">
    ...
    <filesAnalyzed>false</filesAnalyzed>
    ...
</Package>
```

7.9 Package verification code field

7.9.1 Description

This field 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. The metadata for the package verification code field is shown in Table 21.

Table 21 — Metadata for the package verification code field

Attribute	Value
Required	Yes
Cardinality	0..1 if `FilesAnalyzed` (7.8) is `true` or omitted, 0..0 (shall be omitted) if `FilesAnalyzed` is `false`.
Algorithm	(see the algorithm below)
Format	Single line of text with 160-bit binary represented as 40 lowercase hexadecimal digits

Algorithm

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

        append templist with "SHA1(file)/n"
    }
sort templist in ascending order by SHA1 value
filelist = templist with "/n"s removed. /* ordered sequence of SHA1 value
s with no separators */
verificationcode = SHA1(filelist)
```

Where SHA1(file) applies a SHA1 algorithm on the contents of file and returns the result in lowercase hexadecimal digits.

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

7.9.2 Intent

Provide a unique identifier based on the files inside each package, eliminating confusion over which version or modification of a specific package the SPDX document refers to. This field also permits embedding the SPDX document within the package without altering the identifier.

7.9.3 Examples

EXAMPLE 1 Tag: `PackageVerificationCode:` (and optionally `(excludes: FileName)`)

`FileName` is specified in 8.1.

```
PackageVerificationCode: d6a770ba38583ed4bb4525bd96e50461655d2758 (exclud
es: ./package.spdx)
```

EXAMPLE 2 RDF: Properties `spdx:packageVerificationCodeValue`, `spdx:packageVerificationCodeExcludedFile` in class `spdx:PackageVerificationCode` in class `spdx:Package`

```
<Package rdf:about="...">
    <packageVerificationCode>
        <PackageVerificationCode>
            <packageVerificationCodeValue>
                d6a770ba38583ed4bb4525bd96e50461655d2758
            </packageVerificationCodeValue>
            <packageVerificationCodeExcludedFile>
                ./package.spdx
            </packageVerificationCodeExcludedFile>
        </PackageVerificationCode>
    </packageVerificationCode>
</Package>
```

7.10 Package checksum field

7.10.1 Description

Provide an independently reproducible mechanism that permits unique identification of a specific package that correlates to the data in this SPDX document. This identifier enables a recipient to determine if any file in the original package has been changed. If the SPDX document is to be included in a package, this value should not be calculated. The SHA-1 algorithm shall be used to provide the checksum by default. The metadata for the package checksum field is shown in Table 22.

Table 22 — Metadata for the package checksum field

Attribute	Value
Required	No
Cardinality	1..*
Algorithm	Algorithms that can be used: `SHA1`, `SHA224`, `SHA256`, `SHA384`, `SHA512`, `MD2`, `MD4`, `MD5`, `MD6`
Format	There are three components, an algorithm identifier (e.g. `SHA1`), a colon separator `:`, and a bit value represented as lowercase hexadecimal digits (appropriate as output to the algorithm).

7.10.2 Intent

Eliminate confusion over which version or modification of a specific package the SPDX document references by providing a unique identifier of the package.

7.10.3 Examples

EXAMPLE 1 Tag: `PackageChecksum:`

```
PackageChecksum: SHA1: 85ed0817af83a24ad8da68c2b5094de69833983c
```

```
PackageChecksum: SHA256: 11b6d3ee554eedf79299905a98f9b9a04e498210b59f1509
4c916c91d150efcd

PackageChecksum: MD5: 624c1abb3664f4b35547e7c73864ad24
```

EXAMPLE 2 RDF: Properties `spdx:algorithm`, `spdx:checksumValue` in class `spdx:checksum` in class `spdx:Package`

```
<Package rdf:about="...">
    <checksum>
        <Checksum>
            <algorithm rdf:resource="spdx:checksumAlgorithm_sha1"/>
            <checksumValue>85ed0817af83a24ad8da68c2b5094de69833983c
            </checksumValue>
        </Checksum>
    </checksum>
    <checksum>
        <Checksum>
            <algorithm rdf:resource="spdx:checksumAlgorithm_sha256"/>
            <checksumValue>
                11b6d3ee554eedf79299905a98f9b9a04e498210b59f15094c916c91d
150efcd
            </checksumValue>
        </Checksum>
    </checksum>
    <checksum>
        <Checksum>
            <algorithm rdf:resource="spdx:checksumAlgorithm_md5"/>
            <checksumValue>624c1abb3664f4b35547e7c73864ad24</checksumValu
e>
        </Checksum>
    </checksum>
</Package>
```

7.11 Package home page field

7.11.1 Description

Provide a place for the SPDX document creator to record a web site that serves as the package's home page. This link can also be used to reference further information about the package referenced by the SPDX document creator.

Use:

— `NONE` if there is no package home page whatsoever.

— `NOASSERTION` if:

  — the SPDX document creator has attempted to but cannot reach a reasonable objective determination;

  — the SPDX document creator has made no attempt to determine this field; or

  — the SPDX document creator has intentionally provided no information (no meaning should be implied by doing so).

The metadata for the package home page field is shown in Table 23.

Table 23 — Metadata for the package home page field

Attribute	Value
Required	No
Cardinality	1..1
Format	Uniform resource locator | `NONE` | `NOASSERTION`

7.11.2 Intent

Save 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 homepage.

7.11.3 Examples

EXAMPLE 1 Tag: `PackageHomePage:`

```
PackageHomePage: http://ftp.gnu.org/gnu/glibc
```

EXAMPLE 2 RDF: Property `doap:homepage` in class `spdx:Package`

```
<Package rdf:about="...">
    <doap:homepage >http://ftp.gnu.org/gnu/glibc/</doap:homepage>
</Package>
```

This specification uses the prefix `doap:` to refer to the DOAP namespace:

```
http://usefulinc.com/ns/doap#
```

7.12 Source information field

7.12.1 Description

Provide a place for the SPDX document creator to record 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 metadata for the source information field is shown in Table 24.

Table 24 — Metadata for the source information field

Attribute	Value
Required	No
Cardinality	1..1
Format	Free form text that can span multiple lines. In `tag:value` format this is delimited by `<text>...</text>`.

7.12.2 Intent

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

7.12.3 Examples

EXAMPLE 1 Tag: `PackageSourceInfo:`

```
PackageSourceInfo: <text>uses glibc-2_11-branch from git://sourceware.org
/git/glibc.git.</text>
```

EXAMPLE 2 RDF: Property `spdx:sourceInfo` in class `spdx:Package`

```
<Package rdf:about="...">
    ...
    <sourceInfo>uses glibc-2_11-branch from
    git://sourceware.org/git/glibc.git.</sourceInfo>
    ...
</Package>
```

7.13 Concluded license field

7.13.1 Description

Contain the license the SPDX document creator has concluded as governing the package or alternative values, if the governing license cannot be determined.

The options to populate this field are limited to:

- A valid SPDX License Expression as defined in Annex D;
- `NONE`, if the SPDX document creator concludes there is no license available for this package; or
- `NOASSERTION` if:
  - the SPDX document creator has attempted to but cannot reach a reasonable objective determination;
  - the SPDX document creator has made no attempt to determine this field; or

— the SPDX document creator has intentionally provided no information (no meaning should be implied by doing so).

If the Concluded License is not the same as the Declared License (7.15), a written explanation should be provided in the Comments on License field (7.16). With respect to `NOASSERTION`, a written explanation in the Comments on License field (7.16) is preferred.

The metadata for the concluded license field is shown in Table 25.

Table 25 — Metadata for the concluded license field

Attribute	Value
Required	Yes
Cardinality	1..1
Format	`<SPDX License Expression>` | `NONE` | `NOASSERTION` where: `<SPDX License Expression>` is a valid SPDX License Expression as defined in Annex D.

7.13.2 Intent

Here, the intent is for the SPDX document creator to analyze the license information in package, and other objective information, e.g., COPYING file, together with the results from any scanning tools, to arrive at a reasonably objective conclusion as to what license governs the package.

7.13.3 Examples

EXAMPLE 1 Tag: `PackageLicenseConcluded:`

```
PackageLicenseConcluded: LGPL-2.0-only
```

```
PackageLicenseConcluded: (LGPL-2.0-only OR LicenseRef-3)
```

EXAMPLE 2 RDF: Property `spdx:licenseConcluded` in `class spdx:Package`

```
<Package rdf:about="...">
  ...
  <licenseConcluded rdf:resource="http://spdx.org/licenses/LGPL-2.0-only"
/>
  ...
</Package>

<Package rdf:about="...">
  ...
  <licenseConcluded>
     <DisjunctiveLicenseSet>
        <member rdf:resource="http://spdx.org/licenses/LGPL-2.0-only" />
        <member rdf:resource="LicenseRef-3" />
     </DisjunctiveLicenseSet>
  </licenseConcluded>
    ...
</Package>
```

7.14 All licenses information from files field

7.14.1 Description

This field is to contain a list of all licenses found in the package. The relationship between licenses (i.e., conjunctive, disjunctive) is not specified in this field – it is simply a listing of all licenses found.

The options to populate this field are limited to:

- The SPDX License List short form identifier, if a detected license is on the SPDX License List;
- A user defined license reference denoted by `LicenseRef-<idstring>` (for a license not on the SPDX License List);
- `NONE`, if no license information is detected in any of the files; or
- `NOASSERTION`, if:
  - the SPDX document creator has made no attempt to determine this field; or
  - the SPDX document creator has intentionally provided no information (no meaning should be implied by doing so).

The metadata for all license information from files field is shown in Table 26.

Table 26 — Metadata for all license information from files field

Attribute	Value
Required	Yes
Cardinality	0..* if `FilesAnalyzed` (7.8) is `true` or omitted, 0..0 (shall be omitted) if `FilesAnalyzed` is `false`.
Format	`<shortIdentifier>` (Annex A) | ["DocumentRef-"`[idstring]`:]"LicenseRef-"`[idstring]` | `NONE` | `NOASSERTION` where: — "DocumentRef-"`[idstring]` is an optional reference to an external SPDX document as described in 6.6. — `[idstring]` is a unique string containing letters, numbers, `.`, or `–`.

7.14.2 Intent

Here, the intention is to capture all license information detected in the actual files.

7.14.3 Examples

EXAMPLE 1 Tag: `PackageLicenseInfoFromFiles:`

```
PackageLicenseInfoFromFiles: GPL-2.0-only

PackageLicenseInfoFromFiles: LicenseRef-1

PackageLicenseInfoFromFiles: LicenseRef-2
```

EXAMPLE 2 RDF: Property `spdx:licenseInfoFromFiles` in class `spdx:Package`

```
<Package rdf:about="...">
    ...
    <licenseInfoFromFiles rdf:resource
      ="https://spdx.org/licenses/GPL-2.0-only" />
    <licenseInfoFromFiles rdf:resource="#LicenseRef-1" />
    <licenseInfoFromFiles rdf:resource="#LicenseRef-2" />
    ...
</Package>
```

7.15 Declared license field

7.15.1 Description

List the licenses that have been declared by the authors of the package. Any license information that does not originate from the package authors, e.g. license information from a third-party repository, should not be included in this field.

The options to populate this field are limited to:

- A valid SPDX License Expression as defined in Annex D;
- `NONE`, if the package contains no license information whatsoever; or
- `NOASSERTION` if:
    - the SPDX document creator has made no attempt to determine this field; or
    - the SPDX document creator has intentionally provided no information (no meaning should be implied by doing so).

The metadata for the declared license field is shown in Table 27.

Table 27 — Metadata for the declared license field

Attribute	Value
Required	Yes
Cardinality	1..1
Format	`<SPDX License Expression>` | `NONE` | `NOASSERTION` where: • `<SPDX License Expression>` is a valid SPDX License Expression as defined in Annex D.

7.15.2 Intent

This is simply the license identified in text in one or more files (for example COPYING file) in the source code package. This field is not intended to capture license information obtained from an external source, such as the package website. Such information can be included in Concluded License (7.13). This field may have multiple Declared Licenses, if multiple licenses are declared at the package level.

7.15.3 Examples

EXAMPLE 1 Tag: `PackageLicenseDeclared:`

```
PackageLicenseDeclared: LGPL-2.0-only

PackageLicenseDeclared: (LGPL-2.0-only AND LicenseRef-3)
```

EXAMPLE 2 RDF: Property `spdx:licenseDeclared` in class `spdx:Package`

```
<Package rdf:about="...">
  ...
  <licenseDeclared rdf:resource="http://spdx.org/licenses/LGPL-2.0-only"/
>
  ...
</Package>

<Package rdf:about="...">
  ...
  <licenseDeclared>
     <ConjunctiveLicenseSet>
        <member rdf:resource="http://spdx.org/licenses/LGPL-2.0-only"/>
        <member rdf:resource="#LicenseRef-3" />
     </ConjunctiveLicenseSet>
  </licenseDeclared>
    ...
</Package>
```

7.16 Comments on license field

7.16.1 Description

This field provides a place for the SPDX document creator to record any relevant background information or analysis that went in to arriving at the Concluded License for a package. If the Concluded License does not match the Declared License or License Information from Files, this should be explained by the SPDX document creator. It is also preferable to include an explanation here when the Concluded License is NOASSERTION. The metadata for the comments on license field is shown in Table 28.

Table 28 — Metadata for the comments on license field

Attribute	Value
Required	No
Cardinality	1..1
Format	Free form text that can span multiple lines. In tag:value format this is delimited by <text>...</text>.

7.16.2 Intent

Here, the intent is to provide the recipient of the SPDX document with a detailed explanation of how the Concluded License was determined if it does not match the License Information from the files or the source code package, is marked NOASSERTION, or other helpful information relevant to determining the license of the package.

7.16.3 Examples

EXAMPLE 1 Tag: PackageLicenseComments:

```
PackageLicenseComments: <text>The license for this project changed with t
he release of version 1.4. The version of the project included here post-
dates the license change.</text>
```

EXAMPLE 2 RDF: Property spdx:licenseComments in class spdx:Package

```
<Package rdf:about="...">
    ...
    <licenseComments>
        This package has been shipped in source and binary form.
        The binaries were created with gcc 4.5.1 and expect to link to
        compatible system run time libraries.
    </licenseComments>
    ...
</Package>
```

7.17 Copyright text field

7.17.1 Description

Identify the copyright holders of the package, as well as any dates present. This will be a free form text field extracted from package information files. The options to populate this field are limited to:

- Any text related to a copyright notice, even if not complete;
- `NONE` if the package contains no copyright information whatsoever; or
- `NOASSERTION`, if
  - the SPDX document creator has made no attempt to determine this field; or
  - the SPDX document creator has intentionally provided no information (no meaning should be implied by doing so).

The metadata for the copyright text field is shown in Table 29.

Table 29 — Metadata for the copyright text field

Attribute	Value
Required	Yes
Cardinality	1..1
Format	Free form text that can span multiple lines | `NONE` | `NOASSERTION`

7.17.2 Intent

Record any copyright notices for the package.

7.17.3 Examples

EXAMPLE 1 Tag: `PackageCopyrightText:`

In `tag:value` format multiple lines are delimited by `<text>...</text>`.

```
PackageCopyrightText: <text>Copyright 2008-2010 John Smith</text>
```

EXAMPLE 2 RDF: Property `spdx:copyrightText` in class `spdx:Package`

```
<Package rdf:about="...">
    ...
    <copyrightText>Copyright 2008-2010 John Smith</copyrightText>
    ...
</Package>
```

7.18 Package summary description field

7.18.1 Description

This field is a short description of the package. The metadata for the package summary description field is shown in Table 30.

Table 30 — Metadata for the package summary description field

Attribute	Value
Required	No
Cardinality	1..1
Format	Free form text that can span multiple lines.

7.18.2 Intent

Here, the intent is to allow the SPDX document creator to provide concise information about the function or use of the package without having to parse the source code of the actual package.

7.18.3 Examples

EXAMPLE 1 Tag: `PackageSummary:`

In `tag:value` format multiple lines are delimited by `<text>...</text>`.

```
PackageSummary: <text>GNU C library.</text>
```

EXAMPLE 2 RDF: Property `spdx:summary` in class `spdx:Package`

```
<Package rdf:about="...">
    ...
    <summary>GNU C library.</summary>
    ...
</Package>
```

7.19 Package detailed description field

7.19.1 Description

This field is a more detailed description of the package. It may also be extracted from the packages itself. The metadata for the package detailed description field is shown in Table 31.

Table 31 — Metadata for the package detailed description field

Attribute	Value
Required	No
Cardinality	1..1
Format	Free form text than can span multiple lines.

7.19.2 Intent

Here, the intent is to provide recipients of the SPDX document with a detailed technical explanation of the functionality, anticipated use, and anticipated implementation of the package. This field may also include a description of improvements over prior versions of the package.

7.19.3 Examples

EXAMPLE 1 Tag: `PackageDescription:`

In `tag:value` format multiple lines are delimited by `<text>...</text>`.

```
PackageDescription: <text>The GNU C Library defines functions that are sp
ecified by the ISO C standard, as well as additional features specific to
POSIX and other derivatives of the Unix operating system, and extensions
specific to GNU systems.</text>
```

EXAMPLE 2 RDF: Property `spdx:description` in class `spdx:Package`

```
<Package rdf:about="...">
  ...
  <description>
    The GNU C Library defines functions that are specified by the ISO C s
tandard, as well as additional features specific to POSIX and other deriv
atives of the Unix operating system, and extensions specific to GNU syste
ms.
  </description>
    ...
</Package>
```

7.20 Package comment field

7.20.1 Description

This field provides a place for the SPDX document creator to record any general comments about the package being described. The metadata for the package comment field is shown in Table 32.

Table 32 — Metadata for the package comment field

Attribute	Value
Required	No
Cardinality	1..1
Format	Free form text that can span multiple lines.

7.20.2 Intent

Here, the intent is to provide the recipient of the SPDX document with more information determined after careful analysis of a package.

7.20.3 Examples

EXAMPLE 1 Tag: `PackageComment:`

In `tag:value` format multiple lines are delimited by `<text>...</text>`.

```
PackageComment: <text>The package includes several sub-packages; see Rela
tionship information.</text>
```

EXAMPLE 2 RDF: Property `rdfs:comment` in class `spdx:Package`

```
<Package rdf:about="...">
  ...
  <rdfs:comment>
    The package includes several sub-packages; see Relationship
information.
  </rdfs:comment>
  ...
</Package>
```

7.21 External reference field

7.21.1 Description

An External Reference allows a Package to reference an external source of additional information, metadata, enumerations, asset identifiers, or downloadable content believed to be relevant to the Package. The metadata for the external reference field is shown in Table 33.

Table 33 — Metadata for the external reference field

Attribute	Value
Required	No
Cardinality	1..*
Format	`<category> <type> <locator>` where: — `<category>` is `SECURITY` | `PACKAGE-MANAGER` | `PERSISTENT-ID` | `OTHER` — `<type>` is one of the types listed in Annex F. — `<locator>` is the unique string with no spaces necessary to access the package-specific information, metadata, or content within the target location. The format of the locator is subject to constraints defined by the `<type>`.

7.21.2 Intent

To indicate an outside source of information, metadata enumerations, asset identifiers, or content relevant to the Package, such as a structured naming scheme identifying Packages with known security vulnerabilities.

7.21.3 Examples

EXAMPLE 1 Tag: `ExternalRef:`

```
ExternalRef: SECURITY cpe23Type cpe:2.3:a:pivotal_software:spring_framewo
rk:4.1.0:*:*:*:*:*:*:*
```

```
ExternalRef: PERSISTENT-ID swh swh:1:cnt:94a9ed024d3859793618152ea559a168
bbcbb5e2
```

```
ExternalRef: OTHER LocationRef-acmeforge acmecorp/acmenator/4.1.3-alpha
```

EXAMPLE 2 RDF: Property `externalRef` in class `spdx:Package` of type `spdx:ExternalRef`

For a listed location:

```
<spdx:Package rdf:about="...">
    ...
    <spdx:externalRef>
        <spdx:ExternalRef>
            <spdx:referenceCategory rdf:resource
              ="spdx:referenceCategory_packageManager" />
            <spdx:referenceType rdf:resource
              ="http://spdx.org/rdf/refeferences/maven-central" />
            <spdx:referenceLocator>org.apache.commons:commons-lang:3.2.1
            </spdx:referenceLocator>
        </spdx:ExternalRef>
    </spdx:externalRef>
    ...
</spdx:package>
```

For an unlisted location:

```
<spdx:Package rdf:about="...">
    ...
    <spdx:externalRef>
        <spdx:ExternalRef>
            <spdx:referenceCategory rdf:resource="spdx:referenceCategory_
other" />
            <spdx:referenceType rdf:resource="http://spdx.org/spdxdocs/sp
dx-tools-v1.2-3F2504E0-4F89-41D3-9A0C-0305E82...LocationRef-acmeforge" />
            <spdx:referenceLocator>acmecorp/acmenator/4.1.3-alpha</spdx:r
eferenceLocator>
        </spdx:ExternalRef>
    </spdx:externalRef>
    ...
</spdx:package>
```

The referenceType value for a non-listed location consists of the SPDX document namespace (see 6.5) followed by a # and the category as defined in 7.21.

7.22 External reference comment field

7.22.1 Description

To provide human-readable information about the purpose and target of the reference. The metadata for the external reference comment field is shown in Table 34.

Table 34 — Metadata for the external reference comment field

Attribute	Value
Required	No
Cardinality	0..1 for each External Reference (7.21)
Format	Free form text that can span multiple lines. In `tag:value` format this is delimited by `<text>...</text>` and is expected to follow an External Reference (7.21) so that the association can be made.

7.22.2 Intent

To inform a human consumer why the reference exists, what kind of information, content or metadata can be extracted. The target's relationship to artifactOf values of files in the package might need to be explained here. If the reference is BINARY, its relationship to PackageDownloadLocation might need to be explained. If the reference is SOURCE, its relationship to PackageDownloadLocation and SourceInformation might need to be explained.

7.22.3 Examples

EXAMPLE 1 Tag: `ExternalRefComment:`

```
ExternalRefComment: <text>NIST National Vulnerability Database (NVD) desc
ribes security vulnerabilities (CVEs) which affect Vendor Product Version
acmecorp:acmenator:6.6.6.</text>
```

EXAMPLE 2 RDF: Property `rdfs:comment` in class `spdx:ExternalRef`

```
<spdx:Package rdf:about="...">
    ...
    <spdx:externalRef>
        <spdx:ExternalRef>
            <spdx:referenceCategory rdf:resource
              ="spdx:referenceCategory_packageManager" />
            <spdx:referenceType rdf:resource
              ="http://spdx.org/rdf/refeferences/maven-central" />
            <spdx:referenceLocator>org.apache.commons:commons-lang:3.2.1
            </spdx:referenceLocator>
            <rdfs:comment>
              NIST National Vulnerability Database (NVD) describes
              security vulnerabilities (CVEs) which affect Vendor Product
              Version acmecorp:acmenator:6.6.6
            </rdfs:comment>
        </spdx:ExternalRef>
    </spdx:externalRef>
    ...
</spdx:package>
```

7.23 Package attribution text field

7.23.1 Description

This field provides a place for the SPDX document creator to record, at the package level, acknowledgements that might be required to be communicated in some contexts. This is not meant to include the package's actual complete license text (see `PackageLicenseConcluded`, `PackageLicenseDeclared` and `PackageLicenseInfoFromFiles`), and might or might not include copyright notices (see also `PackageCopyrightText`). The SPDX document creator might use this field to record other acknowledgements, such as particular clauses from license texts, which might be necessary or desirable to reproduce. The metadata for the package attribution text field is shown in Table 35.

Table 35 — Metadata for the package attribution text field

Attribute	Value
Required	No
Cardinality	1..*
Format	Free form text that can span multiple lines.

7.23.2 Intent

The intent is to provide the recipient of the SPDX document with acknowledgement content at a package level, to assist redistributors of the package with reproducing those acknowledgements. This field does not necessarily indicate where, or in which contexts, the acknowledgements need to be reproduced (such as end-user documentation, advertising materials, etc.) and the SPDX document creator might or might not explain elsewhere how they intend for this field to be used.

7.23.3 Examples

EXAMPLE 1 Tag: `PackageAttributionText:`

In `tag:value` format multiple lines are delimited by `<text> .. </text>`.

```
PackageAttributionText: <text> All advertising materials mentioning featu
res or use of this software must display the following acknowledgement:
This product includes software developed by the AT&T. </text>
```

EXAMPLE 2 RDF: Property `spdx:attributionText` in class `spdx:Package`

```
<Package rdf:about="...">
    <attributionText>
        All advertising materials mentioning features or use of this
        software must display the following acknowledgement:  This
        product includes software developed by the AT&T.
    </attributionText>
</Package>
```

8 File information section

8.1 File name field

8.1.1 Description

Identify the full path and filename that corresponds to the file information in this section. The metadata for the file name field is shown in Table 36.

Table 36 — Metadata for the file name field

Attribute	Value
Required	Yes
Cardinality	1..1
Format	A relative filename with the root of the package archive or directory. In general, every filename is preceded with a `./`, see http://www.ietf.org/rfc/rfc3986.txt for syntax.

8.1.2 Intent

To aid finding the correct file which corresponds to the file information.

8.1.3 Examples

EXAMPLE 1 Tag: `FileName:`

```
FileName: ./package/foo.c
```

EXAMPLE 2 RDF: Property `spdx:fileName` in class `spdx:File`

```
<File rdf:about="...">
    <fileName>./package/foo.c</fileName>
    ...
</File>
```

8.2 File SPDX identifier field

8.2.1 Description

Uniquely identify any element in an SPDX document which might be referenced by other elements. These might be referenced internally and externally with the addition of the SPDX document identifier. The metadata for the file SPDX identifier field is shown in Table 37.

Table 37 — Metadata for the file SPDX identifier field

Attribute	Value
Required	Yes
Cardinality	1..1
Format	"SPDXRef-"`[idstring]` where `[idstring]` is a unique string containing letters, numbers, `.` and/or −.

8.2.2 Intent

There may be several versions of the same file within an SPDX document. Each element needs to be able to be referred to uniquely so that relationships between elements can be clearly articulated.

8.2.3 Examples

EXAMPLE 1 Tag: `SPDXID:`

```
SPDXID: SPDXRef-1
```

EXAMPLE 2 RDF: The URI for the element will follow the form: [SpdxDocumentURI]#SPDXRef-[idstring] where [SpdxDocumentURI] is the URI for the SPDX document containing the element.

Using `xml:base:`

```
<rdf:RDF xml:base="http://acme.com/spdxdocs/acmeproj/v1.2/1BE2A4FF-5F1A-4
8D3-8483-28A9B0349A1B"
     ...
     <File rdf:about="#SPDXRef-1">
         ...
     </File>
```

Using document URI:

```
<File rdf:about="http://acme.com/spdxdocs/acmeproj/v1.2/1BE2A4FF-5F1A-48D
3-8483-28A9B0349A1B#SPDXRef-1">
     ...
</File>
```

8.3 File type field

8.3.1 Description

This field provides information about the type of file identified. File Type is intrinsic to the file, independent of how the file is being used. A file may have more than one file type assigned to it, however the options to populate this field are limited to:

- `SOURCE` if the file is human readable source code (.c, .html, etc.);
- `BINARY` if the file is a compiled object, target image or binary executable (.o, .a, etc.);
- `ARCHIVE` if the file represents an archive (.tar, .jar, etc.);
- `APPLICATION` if the file is associated with a specific application type (MIME type of application/*);
- `AUDIO` if the file is associated with an audio file (MIME type of audio/* , e.g. .mp3);
- `IMAGE` if the file is associated with a picture image file (MIME type of image/*, e.g., .jpg, .gif);
- `TEXT` if the file is human readable text file (MIME type of text/*);
- `VIDEO` if the file is associated with a video file type (MIME type of video/*);
- `DOCUMENTATION` if the file serves as documentation;
- `SPDX` if the file is an SPDX document;
- `OTHER` if the file doesn't fit into the above categories (generated artifacts, data files, etc.)

The metadata for the file type field is shown in Table 38.

Table 38 — Metadata for the file type field

Attribute	Value
Required	No
Cardinality	0..*
Format	SOURCE | BINARY | ARCHIVE | APPLICATION | AUDIO | IMAGE | TEXT | VIDEO | DOCUMENTATION | SPDX | OTHER

8.3.2 Intent

Here, this field is a reasonable estimation of the file type, from a developer perspective.

8.3.3 Examples

EXAMPLE 1 Tag: `FileType:`

```
FileType: BINARY
```

For a `README.TXT`

```
FileType: TEXT
FileType: DOCUMENTATION
```

foo.exe

```
FileType: BINARY
FileType: APPLICATION
```

EXAMPLE 2 RDF: Property `spdx:fileType` in class `spdx:File`

```
<File rdf:about="file1">
    <fileType rdf:resource="fileType_binary" />
</File>
```

Where file2 is a `README.TXT`

```
<File rdf:about="file2">
    <fileType rdf:resource
      ="http://spdx.org/rdf/terms#fileType_text" />
    <fileType rdf:resource
      ="http://spdx.org/rdf/terms#fileType_documentation" />
</File>
```

8.4 File checksum field

8.4.1 Description

Provide a unique identifier to match analysis information on each specific file in a package. The metadata for the file checksum field is shown in Table 39.

Table 39 — Metadata for the file checksum field

Attribute	Value
Required	Yes
Cardinality	1..1 `SHA1`, others may be optionally provided.
Algorithm	`SHA1` is to be used on the file. Other algorithms that can be provided optionally include `SHA224`, `SHA256`, `SHA384`, `SHA512`, `MD2`, `MD4`, `MD5`, `MD6`
Format	In `tag:value` there are three components, an algorithm identifier (SHA1), a separator (":") and a checksum value. The RDF shall also contain an algorithm identifier and a checksum value. For example, when the algorithm identifier is SHA1, the checksum value should be a 160-bit value represented as 40 lowercase hexadecimal digits. For other algorithms, an appropriate number of hexadecimal digits is expected.

8.4.2 Intent

Here, by providing a unique identifier of each file, confusion over which version/modification of a specific file should be eliminated.

8.4.3 Examples

EXAMPLE 1 Tag: `FileChecksum:`

```
FileChecksum: SHA1: d6a770ba38583ed4bb4525bd96e50461655d2758

FileChecksum: MD5: 624c1abb3664f4b35547e7c73864ad24
```

EXAMPLE 2 RDF: Property `spdx:Checksum` in class `spdx:File`

```
<File rdf:about="...">
    <checksum>
        <Checksum>
            <algorithm rdf:resource
              ="http://spdx.org/rdf/terms#checksumAlgorithm_sha1"/>
            <checksumValue>d6a770ba38583ed4bb4525bd96e50461655d2758
            </checksumValue>
        </Checksum>
    </checksum>
    <checksum>
        <Checksum>
            <algorithm rdf:resource
              ="http://spdx.org/rdf/terms#checksumAlgorithm_md5"/>
            <checksumValue> 624c1abb3664f4b35547e7c73864ad24
            </checksumValue>
        </Checksum>
    </checksum>
</File>
```

8.5 Concluded license field

8.5.1 Description

This field contains the license the SPDX document creator has concluded as governing the file or alternative values if the governing license cannot be determined.

The options to populate this field are limited to:

A valid SPDX License Expression as defined in Annex D;

`NONE`, if the SPDX document creator concludes there is no license available for this file; or

`NOASSERTION`, if:

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

If the Concluded License is not the same as the License Information in File, a written explanation should be provided in the Comments on License field (8.7). With respect to `NOASSERTION`, a written explanation in the Comments on License field (8.7) is preferred. The metadata for the concluded license field is shown in Table 40.

Table 40 — Metadata for the concluded license field

Attribute	Value
Required	Yes
Cardinality	1..1
Format	`<SPDX License Expression>`|`NONE`|`NOASSERTION` where: `<SPDX License Expression>` is a valid SPDX License Expression as defined in Annex D.

8.5.2 Intent

Here, the intent is for the SPDX document creator to analyze the License Information in File (8.6) and other objective information, e.g., "COPYING FILE," along with the results from any scanning tools, to arrive at a reasonably objective conclusion as to what license governs the file.

8.5.3 Examples

EXAMPLE 1 Tag: `LicenseConcluded:`

```
LicenseConcluded: LGPL-2.0-only

LicenseConcluded: (LGPL-2.0-only OR LicenseRef-2)
```

EXAMPLE 2 RDF: Property `spdx:licenseConcluded` in class `spdx:File`

```
<File rdf:about="file">
    <licenseConcluded>LGPL-2.0-only</licenseConcluded>
</File>

<File rdf:about="...">
    <licenseConcluded>
        <DisjunctiveLicenseSet>
            <member rdf:resource
              ="http://spdx.org/licenses/LGPL-2.0-only"/>
            <member rdf:resource="#LicenseRef-2"/>
        </DisjunctiveLicenseSet>
    </licenseConcluded>
</File>
```

8.6 License information in file field

8.6.1 Description

This field contains the license information actually found in the file, if any. This information is most commonly found in the header of the file, although it might be in other areas of the actual file. Any license information not actually in the file, e.g., "COPYING.txt" file in a top-level directory, should not be reflected in this field.

The options to populate this field are limited to:

The SPDX License List short form identifier, if the license is on the SPDX License List; A reference to the license, denoted by LicenseRef-`[idstring]`, if the license is not on the SPDX License List;

`NONE`, if the file contains no license information whatsoever; or

`NOASSERTION`, if:

- the SPDX document creator has made no attempt to determine this field; or
- the SPDX document creator has intentionally provided no information (no meaning should be implied by doing so).

If license information for more than one license is contained in the file or if the license information offers the package recipient a choice of licenses, then each of the choices should be listed as a separate entry. The metadata for the license information in file field is shown in Table 41.

Table 41 — Metadata for the license information in file field

Attribute	Value
Required	Yes
Cardinality	1..*
Format	`<SPDX License Expression>` | ["DocumentRef-"`[idstring]`":"]"LicenseRef-"`[idstring]` | | `NONE` | `NOASSERTION` where: `<SPDX License Expression>` is a valid SPDX License Expression as defined in Annex D. "DocumentRef-"`[idstring]`: is an optional reference to an external SPDX document as described in 6.6 `[idstring]` is a unique string containing letters, numbers, `.` and/or –

8.6.2 Intent

Here, the intent is to provide the license information actually in the file, as compared to the Concluded License field.

8.6.3 Examples

EXAMPLE 1 Tag: `LicenseInfoInFile:`

```
LicenseInfoInFile: GPL-2.0-only
LicenseInfoInFile: LicenseRef-2
```

EXAMPLE 2 RDF: Property `spdx:licenseInfoInFile` in class `spdx:File`

```
<File rdf:about="file1">
    <licenseInfoInFile rdf:resource
      ="http://spdx.org/licenses/GPL-2.0-only" />
    <licenseInfoInFile rdf:resource="#LicenseRef-2" />
</File>
```

8.7 Comments on license field

8.7.1 Description

This field provides a place for the SPDX document creator to record any relevant background references or analysis that went in to arriving at the Concluded License for a file. If the Concluded License does not match the License Information in File, this should be explained by the SPDX document creator. It is also preferable to include an explanation here when the Concluded License is NOASSERTION. The metadata for the comments on license field is shown in Table 42.

Table 42 — Metadata for the comments on license field

Attribute	Value
Required	No
Cardinality	0..1
Format	Free form text that can span multiple lines

8.7.2 Intent

Here, the intent is to provide the recipient of the SPDX document with a detailed explanation of how the Concluded License was determined if it does not match the License Information in File, is marked NOASSERTION, or other helpful information relevant to determining the license of the file.

8.7.3 Examples

EXAMPLE 1 Tag: `LicenseComments:`

In `tag:value` format multiple lines are delimited by `<text> .. </text>`.

```
LicenseComments: <text>The concluded license was taken from the package l
evel that the file was included in. This information was found in the COP
YING.txt file in the xyz directory.</text>
```

EXAMPLE 2 RDF: Property `spdx:licenseComments` in class `spdx:File`

```
<File rdf:about="...">
    <licenseComments>
        The concluded license was taken from the package level that
        the file was included in. This information was found in the
        COPYING.txt file in the xyz directory. This package has been
        shipped in source and binary form.
    </licenseComments>
</File>
```

8.8 Copyright text field

8.8.1 Description

Identify the copyright holder of the file, as well as any dates present. This shall be a free-form text field extracted from the actual file.

The options to populate this field are limited to:

Any text relating to a copyright notice, even if not complete;

NONE, if the file contains no copyright information whatsoever; or

NOASSERTION, if

— the SPDX document creator has made no attempt to determine this field; or

— the SPDX document creator has intentionally provided no information (no meaning should be implied from the absence of an assertion).

The metadata for the copyright text field is shown in Table 43.

Table 43 — Metadata for the copyright text field

Attribute	Value
Required	Yes
Cardinality	1..1
Format	Free form text that can span multiple lines | `NONE` | `NOASSERTION`

8.8.2 Intent

Record any copyright notice for the file.

8.8.3 Examples

EXAMPLE 1 Tag: `FileCopyrightText:`

In `tag:value` format multiple lines are delimited by `<text> .. </text>`.

```
FileCopyrightText: <text> Copyright 2008-2010 John Smith </text>
```

EXAMPLE 2 RDF: Property `spdx:copyrightText` in class `spdx:File`

```
<File rdf:about="...">
    <copyrightText>
        Copyright 2008-2010 John Smith
    </copyrightText>
</File>
```

8.9 Artifact of project name field (deprecated)

8.9.1 Description

To indicate that a file has been derived from a specific project. The metadata for the artifact of project name field is shown in Table 44.

Table 44 — Metadata for the artifact of project name field

Attribute	Value
Required	No
Cardinality	0..*
Format	Single line of text. In `tag:value` format the ArtifactOfProjectName shall precede any optional ArtifactOf optional properties (e.g. ArtifactOfHomePage and ArtifactOfURI).

8.9.2 Intent

To make it easier for recipients of the SPDX document to determine the original source of the identified file. If the project is not described in an SPDX document, then `ArtifactOf` can be used.

If the project is described in another SPDX document, then Relationship should be used.

8.9.3 Examples

EXAMPLE 1 Tag: `ArtifactOfProjectName:`

```
ArtifactOfProjectName: Jena
```

EXAMPLE 2 RDF: Property `doap:name` in class `doap:Project` in property `spdx:artifactOf` in class s`pdx:File`

```
<File>
    <artifactOf>
        <doap:Project>
            <doap:name>Jena</doap:name>
        </doap:Project>
    </artifactOf>
</File>
```

8.10 Artifact of project homepage field (deprecated)

8.10.1 Description

To indicate the location of the project from which the file has been derived. The metadata for the artifact of project homepage field is shown in Table 45.

Table 45 — Metadata for the artifact of project homepage field

Attribute	Value
Required	No
Cardinality	0..*
Format	Uniform Resource Locator | `UNKNOWN`. In `tag:value` format all optional `ArtifactOf` fields shall follow immediately below the ArtifactOfProjectName.

8.10.2 Intent

To make it easier for recipients of the SPDX document to determine the original source of the identified file. If the project is described in another SPDX document, then `Relationship` should be used.

8.10.3 Examples

EXAMPLE 1 Tag: Property `ArtifactOfProjectHomePage` in class `spdx:File`

```
ArtifactOfProjectHomePage: http://www.openjena.org/
```

EXAMPLE 2 RDF: Property `doap:homepage` in class `doap:Project` in property `spdx:artifactOf` in class `spdx:File`

```
<File>
    <artifactOf>
        <doap:Project>
            <doap:homepage >rttp://www.openjena.org/</doap:homepage>
        </doap:Project>
    </artifactOf>
</File>
```

8.11 Artifact of project uniform resource identifier field (deprecated)

8.11.1 Description

To provide a linkage to the project resource in the DOAP document and permit interoperability between the different formats supported. The metadata for the artifact of project uniform resource identifier field is shown in Table 46.

Table 46 — Metadata for the artifact of project uniform resource identifier field

Attribute	Value
Required	No
Cardinality	0..*
Format	Uniform Resource Identifier. In `tag:value` format all optional ArtifactOf fields shall follow immediately below the ArtifactOfProjectName.

In `tag:value` format all optional ArtifactOf fields shall follow immediately below the ArtifactOfProjectName.

8.11.2 Intent

To make it easier for recipients of the SPDX document to determine the original source of the identified file. If the project is described in another SPDX document, then `Relationship` should be used.

8.11.3 Examples

EXAMPLE 1 Tag: `ArtifactOfProjectURI:`

```
ArtifactOfProjectURI: http://subversion.apache.org/doap.rdf
```

EXAMPLE 2 RDF: Property `spdx:artifactOf` in class `spdx:File`

```
<File>
    <artifactOf rdf:resource="http://subversion.apache.org/" />
</File>
```

```
<!-- Note: within the DOAP file at http://subversion.apache.org/doap.rdf
the value "http://subversion.apache.org/" is the URI of the describes
resource of type doap:Project -->
```

8.12 File comment field

8.12.1 Description

This field provides a place for the SPDX document creator to record any general comments about the file. The metadata for the file comment field is shown in Table 47.

Table 47 — Metadata for the file comment field

Attribute	Value
Required	No
Cardinality	0..1
Format	Free form text that may span multiple lines

8.12.2 Intent

Here, the intent is to provide the recipient of the SPDX document with more information determined after careful analysis of a file.

8.12.3 Examples

EXAMPLE 1 Tag: `FileComment:`

In `tag:value` format multiple lines are delimited by `<text> .. </text>`.

```
FileComment: <text>
This file appears in other packages, such as Foo and Ufoo.
</text>
```

EXAMPLE 2 RDF: Property `rdfs:comments` in class `spdx:File`

```
<File rdf:about="...">
    <rdfs:comment>
        This file appears in other packages, such as Foo and Ufoo.
    </rdfs:comment>
</File>
```

8.13 File notice field

8.13.1 Description

This field provides a place for the SPDX document creator to record license notices or other such related notices found in the file. This might or might not include copyright statements. The metadata for the file notice field is shown in Table 48.

Table 48 — Metadata for the file notice field

Attribute	Value
Required	No
Cardinality	0..1
Format	Free form text that can span multiple lines

8.13.2 Intent

Here, the intent is to provide the recipient of the SPDX document with notices that may require additional review or otherwise contribute to the determination of the Concluded License.

8.13.3 Examples

EXAMPLE 1 Tag: `FileNotice:`

In `tag:value` format multiple lines are delimited by `<text> .. </text>`.

```
FileNotice: <text>This file is licensed under GPL.</text>
```

EXAMPLE 2 RDF: Property `noticeText` in class `spdx:File`

```
<File rdf:about="...">
    <noticeText>
        This file is licensed under GPL.
    </noticeText>
</File>
```

8.14 File contributor field

8.14.1 Description

This field provides a place for the SPDX document creator to record file contributors. Contributors could include names of copyright holders and/or authors who might not be copyright holders, yet contributed to the file content. The metadata for the file contributor field is shown in Table 49.

Table 49 — Metadata for the file contributor field

Attribute	Value
Required	No
Cardinality	0..*
Format	Free form text on a single line.

8.14.2 Intent

Here, the intent is to provide the recipient of the SPDX document with a list of one or more contributors (credits). This is one way of providing acknowledgement to the contributors of a file. This would be useful, for example, if a recipient company wanted to contact copyright holders to inquire about alternate licensing.

8.14.3 Examples

EXAMPLE 1 Tag: `FileContributor:`

In `tag:value` format single line per contributor.

```
FileContributor: Modified by Paul Mundt lethal@linux-sh.org
FileContributor: The Regents of the University of California
FileContributor: IBM Corporation
```

EXAMPLE 2 RDF: Property `spdx:fileContributor` in class `spdx:File`

```
<File rdf:about="...">
    <fileContributor> Modified by Paul Mundt lethal@linux-sh.org
    </fileContributor>
    <fileContributor> The Regents of the University of California
    </fileContributor>
    <fileContributor> IBM Corporation </fileContributor>
</File>
```

8.15 File attribution text field

8.15.1 Description

This field provides a place for the SPDX document creator to record, at the file level, acknowledgements that might be required to be communicated in some contexts. This is not meant to include the file's actual complete license text (see `LicenseConcluded` and `LicenseInfoInFile`), and might or might not include copyright notices (see also `FileCopyrightText`). The SPDX document creator might use this field to record other acknowledgements, such as particular clauses from license texts, which might be necessary or desirable to reproduce. The metadata for the file attribution text field is shown in Table 50.

Table 50 — Metadata for the file attribution text field

Attribute	Value
Required	No
Cardinality	0..*
Format	Free form text that can span multiple lines.

8.15.2 Intent

The intent is to provide the recipient of the SPDX document with acknowledgement content at a file level, to assist redistributors of the file with reproducing those acknowledgements. This field does not necessarily indicate where, or in which contexts, the acknowledgements need to be reproduced (such as end-user documentation, advertising materials, etc.) and the SPDX document creator might or might not explain elsewhere how they intend for this field to be used.

8.15.3 Examples

EXAMPLE 1 Tag: `FileAttributionText:`

In `tag:value` format multiple lines are delimited by `<text> .. </text>`.

```
FileAttributionText: <text> All advertising materials mentioning features
or use of this software must display the following acknowledgement:  This
product includes software developed by the AT&T. </text>
```

EXAMPLE 2 RDF: Property `spdx:attributionText` in class `spdx:File`

```
<File rdf:about="...">
    <attributionText>
        All advertising materials mentioning features or use of
        this software must display the following acknowledgement:
        This product includes software developed by the AT&T.
    </attributionText>
</File>
```

8.16 File dependencies field (deprecated)

This field is deprecated since SPDX 2.0 in favor of using Clause 11 which provides more granularity about relationships.

8.16.1 Description

The field provides a place for the SPDX document creator to record a list of other files (referenceable within this SPDX document) which the file is a derivative of and/or depends on for the build (e.g., source file or build script for a binary program or library). The list of files might not necessarily represent the list of all file dependencies, but possibly the ones that impact the licensing and/or might be needed as part of the file distribution obligation. The metadata for the file dependencies field is shown in Table 51.

Table 51 — Metadata for the file dependencies field

Attribute	Value
Required	No
Cardinality	0..*
Format	Reference to the file within the SPDX document. For the `tag:value` format, this will be the filename. For the RDF format, it shall be a reference to the actual file node.

8.16.2 Intent

Here, the intent is to provide the recipient of the SPDX document with file dependency information based on the build system that created the file. These other files might impact the licensing of the file and/or might be required to satisfy the distribution obligation of the file (e.g., source files subject to a copyleft license).

8.16.3 Examples

EXAMPLE 1 Tag: `FileDependency:`

```
FileDependency: ./busybox-1.20.2/shell/match.h
FileDependency: ./busybox-1.20.2/shell/match.c
FileDependency: ./busybox-1.20.2/shell/ash.c
```

EXAMPLE 2 RDF: Property `spdx:fileDependency` in class `spdx:File`

```
<File rdf:nodeID="A0">
    <fileName>./package/source1.java</fileName>
</File>

<File rdf:nodeID="A1">
    <fileName>./package/source2.java</fileName>
</File>

<File rdf:nodeID="A3">
  <fileName>./package/source3.java</fileName>
</File>

<File rdf:about="...">
    <fileName>./package/mylibrary.jar</fileName>
    <fileDependency rdf:nodeID="A0"/>
    <fileDependency rdf:nodeID="A1"/>
    <fileDependency rdf:nodeID="A2"/>
</File>
```

9 Snippet information section

9.1 Snippet SPDX identifier field

9.1.1 Description

Uniquely identify any element in an SPDX document which may be referenced by other elements. These may be referenced internally and externally with the addition of the SPDX document identifier. The metadata for the snippet SPDX identifier field is shown in Table 52.

Table 52 — Metadata for the snippet SPDX identifier field

Attribute	Value
Required	Yes
Cardinality	1..1
Format	`SPDXRef-[idstring]` where `[idstring]` is a unique string containing letters, numbers, . and/or -.

9.1.2 Intent

There may be several instances of a snippet within an SPDX document. Each snippet is an element which needs to be able to be referred to uniquely so that relationships between it and other elements can be clearly articulated.

9.1.3 Examples

EXAMPLE 1 Tag: `SnippetSPDXID:`

```
SnippetSPDXID: SPDXRef-1
```

EXAMPLE 2 RDF:

The URI for the element shall follow the form: `[SpdxDocumentURI]#SPDXRef-[idstring]` where `[SpdxDocumentURI]` is the URI for the SPDX document containing the element.

Using xml:base:

```
<rdf:RDF xml:base="http://acme.com/spdxdocs/acmeproj/v1.2/1BE2A4FF-5F1A-4
8D3-8483-28A9B0349A1B"
    ...
    <Snippet rdf:about="#SPDXRef-1">
        ...
    </Snippet>
```

Using document URI:

```
<Snippet rdf:about="http://acme.com/spdxdocs/acmeproj/v1.2/1BE2A4FF-5F1A-
48D3-8483-28A9B0349A1B#SPDXRef-1">
    ...
</Snippet>
```

9.2 Snippet from file SPDX identifier field

9.2.1 Description

Uniquely identify the file in an SPDX document which this snippet is associated with. The metadata for the snippet from file SPDX identifier field is shown in Table 53.

Table 53 — Metadata for the snippet from file SPDX identifier field

Attribute	Value
Required	Yes
Cardinality	1..1
Format	["DocumentRef-"[idstring]":"] SPDXID where `DocumentRef-[idstring]`: is an optional reference to an external SPDX document as described in 6.6 where `SPDXID` is a string containing letters, numbers, `.` and/or `-`. as described in (6.3, 7.2, 8.2).

9.2.2 Intent

There may be several versions of the same file within an SPDX document. Each element needs to be able to be referred to uniquely so that relationships between elements can be clearly articulated.

9.2.3 Examples

EXAMPLE 1 Tag: `SnippetFromFileSPDXID:`

Snippet from a File in local SPDX Doc:

```
SnippetFromFileSPDXID: SPDXRef-filecontainingsnippet
```

Snippet from a File in an External SPDX Doc:

```
SnippetFromFileSPDXID: DocumentRef-ExternalDoc1:SPDXRef-filecontainingsni
ppet
```

EXAMPLE 2 RDF: Property `spdx:snippetFromFile` in class `spdx:Snippet`

Snippet from a File in local SPDX Doc:

```
<Snippet rdf:ID="SPDXRef-1">
    <snippetFromFile rdf:about="#SPDXRef-filecontainingsnippet">
        ...
    </Snippet>
```

Snippet from a File in an External SPDX Doc:

```
<Snippet rdf:ID="SPDXRef-1">
    <snippetFromFile rdf:about="http://foo.org/ExternalDocument1#SPDXRef-
filecontainingsnippet">
    ...
</Snippet>
```

9.3 Snippet byte range field

9.3.1 Description

This field defines the byte range in the original host file (in 9.2) that the snippet information applies to. The metadata for the snippet byte range field is shown in Table 54.

Table 54 — Metadata for the snippet byte range field

Attribute	Value
Required	Yes
Cardinality	1..1
Format	`number1:number2` where: `number1` is greater than or equal to 1 and less or equal to `number2`, AND `number2` is less than or equal to the total number of bytes in file. The byte at position number1 and position number2 are included in the range.

9.3.2 Intent

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 (see http://www.w3.org/TR/Pointers-in-RDF10/).

9.3.3 Examples

EXAMPLE 1 Tag: `SnippetByteRange:`

```
SnippetByteRange: 310:420
```

EXAMPLE 2 RDF: Property `spdx:Range` in class `spdx:Snippet`.

The RDF uses the W3C proposed pointer method vocabulary (see http://www.w3.org/TR/Pointers-in-RDF10/

Supported classes from the pointer method vocabulary are `StartEndPointer` and `ByteOffsetPointer`. Supported properties from the pointer method vocabulary include:

- `startPointer`
- `endPointer`
- `reference`
- `offset`

```
<Snippet rdf:about="...">
    <range>
        <ptr:StartEndPointer>
            <ptr:startPointer>
```

```
                <ptr:ByteOffsetPointer>
                    <ptr:reference rdf:resource="#SPDXRef-fileReference/>
                    <ptr:offset>310</ptr:offset>
                </ptr:ByteOffsetPointer>
            </ptr:startPointer>
            <ptr:endPointer>
                <ptr:ByteOffsetPointer>
                    <ptr:reference  rdf:resource="#SPDXRef-fileReference/
>

                    <ptr:offset>420</ptr:offset>
                </ptr:ByteOffsetPointer>
            </ptr:endPointer>
        </ptr: StartEndPointer>
    </range>
</Snippet>
```

This specification uses the prefix `ptr:` to refer to the W3C Pointers namespace:

`xmlns:ptr=http://www.w3.org/2009/pointers#`

9.4 Snippet line range field

9.4.1 Description

This optional field defines the line range in the original host file (see 9.2) 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. The metadata for the snippet line range field is shown in Table 55.

Table 55 — Metadata for the snippet line range field

Attribute	Value
Required	No
Cardinality	0..1
Format	`number1:number2` where: `number1` is greater than or equal to 1 and less than or equal to `number2`, AND `number2` is less than or equal to the total number of lines in file.

9.4.2 Intent

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 (see http://www.w3.org/TR/Pointers-in-RDF10/).

9.4.3 Examples

EXAMPLE 1 Tag: `SnippetLineRange:`

```
SnippetLineRange: 5:23
```

EXAMPLE 2 RDF: Properties `spdx:Range` in class `spdx:Snippet`.

The RDF uses the W3C proposed pointer method vocabulary (see http://www.w3.org/TR/Pointers-in-RDF10/).

Supported classes from the pointer method vocabulary are `StartEndPointer` and `LineCharPointer`. Supported properties from the pointer method vocabulary include:

— `startPointer`

— `endPointer`

— `reference`

— `lineNumber`

```
<Snippet rdf:about="...">
    <range>
        <ptr:StartEndPointer>
            <ptr:startPointer>
                <ptr:LineCharPointer>
                    <ptr:reference rdf:resource="#SPDXRef-fileReference"/
>
                    <ptr:lineNumber>5</ptr:lineNumber>
                </ptr:LineCharPointer>
            </ptr:startPointer>
            <ptr:endPointer>
            <ptr:LineCharPointer>
                <ptr:reference rdf:resource="#SPDXRef-fileReference"/>
                <ptr:lineNumber>23</ptr:lineNumber>
            </ptr:LineCharPointer>
        </ptr: StartEndPointer>
    </range>
</Snippet>
```

9.5 Snippet concluded license field

9.5.1 Description

This field contains the license the SPDX document creator has concluded as governing the snippet or alternative values if the governing license cannot be determined. The options to populate this field are limited to:

A valid SPDX License Expression as defined in Annex D.

`NONE` should be used if there is no licensing information from which to conclude a license for the snippet.

`NOASSERTION` should be used if for the snippet:

— the SPDX document creator has attempted to, but cannot reach a reasonable objective determination of the Concluded License;

— the SPDX document creator is uncomfortable concluding a license, despite some license information being available;

— the SPDX document creator has made no attempt to determine a Concluded License;

— the SPDX document creator has intentionally provided no information (no meaning should be implied by doing so).

If the Concluded License is not the same as the License Information in Snippet, a written explanation should be provided in the Comments on License field (see 9.7). With respect to `NOASSERTION`, a written explanation in the Comments on License field (see 9.7) is preferred. The metadata for the snippet concluded license field is shown in Table 56.

Table 56 — Metadata for the snippet concluded license field

Attribute	Value
Required	Yes
Cardinality	1..1
Format	`<SPDX License Expression>` | `NONE` | `NOASSERTION` where: `<SPDX License Expression>` is a valid SPDX License Expression as defined in Annex D.

9.5.2 Intent

Here, the intent is for the SPDX document creator to reconcile the license information known about the snippet, what license information is in the snippet itself and other objective information for a package, along with the results from any scanning tools, to arrive at a reasonably objective conclusion as to what license governs the snippet.

9.5.3 Examples

EXAMPLE 1 Tag: `SnippetLicenseConcluded:`

```
SnippetLicenseConcluded: GPL-2.0-only

SnippetLicenseConcluded: (LGPL-2.0-only OR LicenseRef-2)
```

EXAMPLE 2 RDF: Property `spdx:licenseConcluded` in class `spdx:Snippet`

```
<Snippet rdf:about="...">
    ...
    <licenseConcluded>GPL-2.0-only</licenseConcluded>
    ...
</Snippet>

<Snippet rdf:about="...">
    <licenseConcluded>
       <DisjunctiveLicenseSet>
         <member rdf:resource="http://spdx.org/licenses/LGPL-2.0-only"/>
         <member rdf:resource="#LicenseRef-2"/>
       </DisjunctiveLicenseSet>
    </licenseConcluded>
</Snippet>
```

9.6 License information in snippet field

9.6.1 Description

This field contains the license information actually found in the snippet, if any. Any license information not actually in the snippet itself, e.g., header of the file the snippet belongs in, "COPYING.txt" file in a top level directory, should not be reflected in this field.

The options to populate this field are limited to:

The SPDX License List short form identifier, if the license is on the SPDX License List; A reference to the license, denoted by LicenseRef-`[idstring]`, if the license is not on the SPDX License List;

`NONE`, if the snippet contains no license information whatsoever; or

`NOASSERTION`, if:

— the SPDX document creator has made no attempt to determine this field; or

— the SPDX document creator has intentionally provided no information (no meaning should be implied by doing so).

If license information for more than one license is contained in the snippet or if the license information offers a choice of licenses, then each of the choices should be listed as a separate entry. The metadata for the license information in snippet field is shown in Table 57.

Table 57 — Metadata for the license information in snippet field

Attribute	Value
Required	No
Cardinality	0..*
Format	`<SPDX License Expression>` | ["DocumentRef-"`[idstring]`:"]"LicenseRef-"[idstring] | | `NONE` | `NOASSERTION` where: `<SPDX License Expression>` is a valid SPDX License Expression as defined in Annex D. DocumentRef-"`[idstring]`: is an optional reference to an external SPDX document as described in 6.6 `[idstring]` is a unique string containing letters, numbers, `.` and/or –.

9.6.2 Intent

Here, the intent is to provide the license information actually in the snippet, as compared to the Concluded License field.

9.6.3 Examples

EXAMPLE 1 Tag: `LicenseInfoInSnippet:`

```
LicenseInfoInSnippet: LGPL-2.0-only

LicenseInfoInSnippet: LicenseRef-2
```

EXAMPLE 2 RDF: Property `spdx:licenseInfoInSnippet` in class `spdx:Snippet`

```
<Snippet rdf:about="...">
    <licenseInfoInSnippet rdf:resource
      ="http://spdx.org/licenses/GPL-2.0-only" />
    <licenseInfoInSnippet rdf:resource="#LicenseRef-2" />
</Snippet>
```

9.7 Snippet comments on license field

9.7.1 Description

This field provides a place for the SPDX document creator to record any relevant background references or analysis that went in to arriving at the Concluded License for a snippet. The metadata for the snippet comments on license field is shown in Table 58.

Table 58 — Metadata for the snippet comments on license field

Attribute	Value
Required	No
Cardinality	0..1
Format	Free form text that can span multiple lines

9.7.2 Intent

Here, the intent is to provide the recipient of the SPDX document with a detailed explanation of how the Concluded License was determined for a Snippet if it does not match the License Information in Snippet, is marked `NOASSERTION`, or other helpful information relevant to determining the license of the snippet in a file.

9.7.3 Examples

EXAMPLE 1 Tag: `SnippetLicenseComments:`

In `tag:value` format multiple lines are delimited by `<text> .. </text>`.

```
SnippetLicenseComments: <text>The concluded license was taken from packag
e xyz, from which the snippet was copied into the current file.
The concluded license information was found in the COPYING.txt file in pa
ckage xyz.</text>
```

EXAMPLE 2 RDF: Property `spdx:licenseComments` in class `spdx:Snippet`

```
<Snippet rdf:about="...">
    ...
    <licenseComments> The concluded license was taken from package xyz, f
rom which the snippet was copied into the current file. The concluded lic
ense information was found in the COPYING.txt file in package xyz. </lice
nseComments>
    ...
</Snippet>
```

9.8 Snippet copyright text field

9.8.1 Description

Identify the copyright holder of the snippet, as well as any dates present. This shall be a free form text field, ideally extracted from the actual snippet. The options to populate this field are limited to:

any text relating to a copyright notice, even if not complete;

`NONE`, if the snippet contains no copyright information whatsoever; or

`NOASSERTION`, if the SPDX document creator has not examined the contents of the actual snippet or if the SPDX document creator has intentionally provided no information (no meaning should be implied from the absence of an assertion).

The metadata for the snippet copyright text field is shown in Table 59.

Table 59 — Metadata for the snippet copyright text field

Attribute	Value
Required	Yes
Cardinality	1..1
Format	Free form text that can span multiple lines | `NONE` | `NOASSERTION`

9.8.2 Intent

Record any copyright notice associated with the snippet.

9.8.3 Examples

EXAMPLE 1 Tag: `SnippetCopyrightText:`

In `tag:value` format multiple lines are delimited by `<text> .. </text>`.

```
SnippetCopyrightText: <text> Copyright 2008-2010 John Smith </text>
```

EXAMPLE 2 RDF: Property `spdx:copyrightText` in class `spdx:Snippet`

```
<Snippet rdf:about="...">
    ...
    <copyrightText>
        Copyright 2008-2010 John Smith
    </copyrightText>
    ...
</Snippet>
```

9.9 Snippet comment field

9.9.1 Description

This field provides a place for the SPDX document creator to record any general comments about the snippet. The metadata for the snippet comment field is shown in Table 60.

Table 60 — Metadata for the snippet comment field

Attribute	Value
Required	No
Cardinality	0..1
Format	Free form text that can span multiple lines

9.9.2 Intent

Here, the intent is to provide the recipient of the SPDX document with more information determined after careful analysis of a snippet.

9.9.3 Examples

EXAMPLE 1 Tag: `SnippetComment:`

In `tag:value` format multiple lines are delimited by `<text> .. </text>`.

```
SnippetComment: <text>This snippet was identified as significant and high
lighted in this Apache-2.0 file, when a commercial scanner identified it
as being derived from file foo.c in package xyz which is licensed under G
PL-2.0.</text>
```

EXAMPLE 2 RDF: Property `rdfs:comment` in class `spdx:Snippet`

```
<Snippet rdf:about="...">
    ...
    <rdfs:comment>
        This snippet was identified as significant and highlighted
        in this Apache-2.0 file, when a commercial scanner identified
        it as being derived from file foo.c in package xyz which is
        licensed under GPL-2.0.
    </rdfs:comment>
    ...
</Snippet>
```

9.10 Snippet name field

9.10.1 Description

Identify a specific snippet in a human convenient manner. The metadata for the snippet name field is shown in Table 61.

Table 61 — Metadata for the snippet name field

Attribute	Value
Required	No
Cardinality	0..1
Format	Single line of text

9.10.2 Intent

To aid in identifying a snippet under discussion that might be used in multiple locations, and for consistency with the ability to refer to any copyrightable SPDX Element by name.

9.10.3 Examples

EXAMPLE 1 Tag: `SnippetName:`

```
SnippetName: from Linux kernel
```

EXAMPLE 2 RDF: Property `spdx:name` in class `spdx:Snippet`

```
<Snippet rdf:about="...">
    <name>from Linux kernel</name>
</Snippet>
```

9.11 Snippet attribution text field

9.11.1 Description

This field provides a place for the SPDX document creator to record, at the snippet level, acknowledgements that may be required to be communicated in some contexts. This is not meant to include the snippet's actual complete license text (see `SnippetLicenseConcluded` and `LicenseInfoInSnippet`), and might or might not include copyright notices (see also `SnippetCopyrightText`). The SPDX document creator may use this field to record other acknowledgements, such as particular clauses from license texts, which might be necessary or desirable to reproduce. The metadata for the snippet attribution text field is shown in Table 62.

Table 62 — Metadata for the snippet attribution text field

Attribute	Value
Required	No
Cardinality	0..*
Format	Free form text that can span multiple lines.

9.11.2 Intent

The intent is to provide the recipient of the SPDX document with acknowledgement content at a snippet level, to assist redistributors of the file with reproducing those acknowledgements. This field does not necessarily indicate where, or in which contexts, the acknowledgements need to be reproduced (such as end-user documentation, advertising materials, etc.) and the SPDX document creator might or might not explain elsewhere how they intend for this field to be used.

9.11.3 Examples

EXAMPLE 1 Tag: `SnippetAttributionText:`

In `tag:value` format multiple lines are delimited by `<text> .. </text>`.

```
SnippetAttributionText: <text> All advertising materials mentioning featu
res or use of this software must display the following acknowledgement:
This product includes software developed by the AT&T. </text>
```

EXAMPLE 2 RDF: Property `spdx:attributionText` in class `spdx:Snippet`

```
<Snippet rdf:about="...">
    <attributionText>
        All advertising materials mentioning features or use of this
        software must display the following acknowledgement:  This
        product includes software developed by the AT&T.
    </attributionText>
</Snippet>
```

10 Other licensing information detected section

10.1 License identifier field

10.1.1 Description

Provide a locally unique identifier to refer to licenses that are not found on the SPDX License List. This unique identifier can then be used in the packages, files and snippets sections of the SPDX document (Clause 7, Clause 8 and Clause 9, respectively).

Table 63 — Metadata for the license identifier field

Attribute	Value
Required	Conditional
Cardinality	0..1 conditional (mandatory, one) if license is not on SPDX License List.
Format	"LicenseRef-"`[idstring]` Where `[idstring]` is a unique string containing letters, numbers, `.` and/or `-`.

10.1.2 Intent

Create a human readable short form license identifier for a license not on the SPDX License List. This identifier shall be unique within the SPDX document. In previous versions of SPDX, the references were required to be sequential numbers, but as of version 1.2, creators may specify references that are easier for humans to remember and mentally map.

10.1.3 Examples

EXAMPLE 1 Tag: `LicenseID:`

```
LicenseID: LicenseRef-1

LicenseID: LicenseRef-Beerware-4.2
```

EXAMPLE 2 RDF: Property `spdx:licenseID` in class `spdx:ExtractedLicensingInfo`

```
<ExtractedLicensingInfo rdf:about="licenseRef-1">
   <licenseId>LicenseRef-1</licenseId>
</ExtractedLicensingInfo>

<ExtractedLicensingInfo rdf:about="licenseRef-Beerware-4.2">
    <licenseId>LicenseRef-Beerware-4.2</licenseId>
</ExtractedLicensingInfo>
```

10.2 Extracted text field

10.2.1 Description

Provide a copy of the actual text of the license reference extracted from the package, file or snippet that is associated with the License Identifier to aid in future analysis. The metadata for the extracted text field is shown in Table 64.

Table 64 — Metadata for the extracted text field

Attribute	Value
Required	Conditional
Cardinality	0..1 conditional (Mandatory, one) if there is a License Identifier assigned.
Format	Free form text field that may span multiple lines.

10.2.2 Intent

Provide the actual text as found in the package, file or snippet for a license that is not on the SPDX License List.

10.2.3 Examples

EXAMPLE 1 Tag: `ExtractedText:`

In `tag:value` format multiple lines are delimited by `<text> .. </text>`.

If only short reference to license present in File:

```
ExtractedText: <text>This software is licensed under the Beer License.</t
ext>
```

If indeed full text of license present in File:

```
ExtractedText: <text>"THE WHISKEY-WARE LICENSE": whiskeyfan@example.com w
rote this file. As long as you retain this notice you can do whatever you
want with this stuff. If we meet some day, and you think this stuff is wo
rth it, you can buy me a bottle of whiskey in return </text>
```

EXAMPLE 2 RDF: Property `spdx:extractedText` in class `spdx:ExtractedLicensingInfo`

If only short reference to license present in File:

```
<ExtractedLicensingInfo rdf:about="licenseRef-Whiskeyware">
    <licenseId>LicenseRef-Whiskeyware</licenseId>
    <extractedText>This software is licensed under the WHISKEY-WARE LICEN
SE.</extractedText>
</ExtractedLicensingInfo>
```

If indeed full text of license present in File:

```
<ExtractedLicensingInfo rdf:about="licenseRef-Whiskeyware">
    <licenseId>LicenseRef-Whiskeyware</licenseId>
    <extractedText>""THE WHISKEY-WARE LICENSE": whiskeyfan@example.com wr
ote this file. As long as you retain this notice you can do whatever you
want with this stuff. If we meet some day, and you think this stuff is wo
rth it, you can buy me a bottle of whiskey in return.</extractedText>
</ExtractedLicensingInfo>
```

10.3 License name field

10.3.1 Description

Provide a common name of the license that is not on the SPDX list.

Use `NOASSERTION` If there is no common name or it is not known. The metadata for the license name field is shown in Table 65.

Table 65 — Metadata for the license name field

Attribute	Value
Required	Conditional
Cardinality	0..1 conditional (mandatory, one) if license is not on SPDX License List.
Format	Single line of text

10.3.2 Intent

Provides a human readable name suitable for use as a title or label of the license when showing compact lists of licenses from the SPDX document to humans.

10.3.3 Examples

EXAMPLE 1 Tag: `LicenseName:`

```
LicenseName: Whiskey-Ware License
```

EXAMPLE 2 RDF: Property `spdx:name` in class `spdx:ExtractedLicensingInfo`

```
<ExtractedLicensingInfo rdf:about="licenseRef-Whiskey-Ware">
   <name>Whiskey-Ware License </name>
</ExtractedLicensingInfo>
```

10.4 License cross reference field

10.4.1 Description

Provide a pointer to the official source of a license that is not included in the SPDX License List, that is referenced by the License Identifier. The metadata for the license cross reference field is shown in Table 66.

Table 66 — Metadata for the license cross reference field

Attribute	Value
Required	Conditional
Cardinality	0..* conditional (optional, one or more) if license is not on SPDX License List.
Format	Uniform Resource Locator

10.4.2 Intent

Canonical source for a license currently not on the SPDX License List.

10.4.3 Examples

EXAMPLE 1 Tag: `LicenseCrossReference:`

```
LicenseCrossReference: http://people.freebsd.org/~phk/
```

EXAMPLE 2 RDF: Property `rdfs:seeAlso` in class `spdx:ExtractedLicensingInfo`

```
<ExtractedLicensingInfo rdf:about="licenseRef-1">
    <rdfs:seeAlso>http://people.freebsd.org/~phk/</rdfs:seeAlso>
</ExtractedLicensingInfo>
```

10.5 License comment field

10.5.1 Description

This field provides a place for the SPDX document creator to record any general comments about the license. The metadata for the license comment field is shown in Table 67.

Table 67 — Metadata for the license comment field

Attribute	Value
Required	No
Cardinality	0..1
Format	Free form text that can span multiple lines

10.5.2 Intent

Here, the intent is to provide the recipient of the SPDX document with more information determined after careful analysis of a license, or addition cross references.

10.5.3 Examples

EXAMPLE 1 Tag: `LicenseComment:`

In `tag:value` format multiple lines are delimited by `<text> .. </text>`.

```
LicenseComment: <text>The Whiskey-Ware License has a couple of other stan
dard variants.</text>
```

EXAMPLE 2 RDF: Property `rdfs:comment` in class `spdx:ExtractedLicensingInfo`

```
<ExtractedLicensingInfo rdf:about="licenseRef-1">
    <rdfs:comment> The Whiskey-Ware License has a couple of other standar
d variants.</rdfs:comment>
</ExtractedLicensingInfo>
```

11 Relationships between SPDX elements information section

11.1 Relationship field

11.1.1 Description

This field provides information about the relationship between two SPDX 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.

In cases where there are "known unknowns", the use of the keyword `NOASSERTION` can be used on the right hand side of a relationship to indicate that the author is not asserting whether there are other SPDX elements (package/file/snippet) that are connected by relationships or not. That is, there could be some, but the author is not asserting one way or another.

Similarly, the use of the keyword `NONE` can be used to indicate that an SPDX element (package/file/snippet) has no other elements connected by some relationship to it.

The use of `NOASSERTION`or `NONE` is not mandatory for any relationship. If no relationship of a particular type is specified, then the document author is not presumed to be asserting whether or not there are relationships of that type. If some relationships of a particular type are specified, then the document author is not presumed to be asserting whether there are more possible relationships of that type.

The relationships between two SPDX elements that are supported are shown in Table 68.

Table 68 — Relationships between two SPDX elements that are supported

Relationship	Description	Example
DESCRIBES	Is to be used when SPDXRef-DOCUMENT describes SPDXRef-A.	An SPDX document `WildFly.spdx` describes package ‘WildFly’. Note this is a logical relationship to help organize related items within an SPDX document that is mandatory if more than one package or set of files (not in a package) is present.
DESCRIBED_BY	Is to be used when SPDXRef-A is described by SPDXREF-Document.	The package ‘WildFly’ is described by SPDX document `WildFly.spdx`.

Relationship	Description	Example
CONTAINS	Is to be used when SPDXRef-A contains SPDXRef-B.	An ARCHIVE file `bar.tgz` contains a SOURCE file `foo.c`.
CONTAINED_BY	Is to be used when SPDXRef-A is contained by SPDXRef-B.	A SOURCE file `foo.c` is contained by ARCHIVE file `bar.tgz`
DEPENDS_ON	Is to be used when SPDXRef-A depends on SPDXRef-B.	Package A depends on the presence of package B in order to build and run
DEPENDENCY_OF	Is to be used when SPDXRef-A is dependency of SPDXRef-B.	A is explicitly stated as a dependency of B in a machine-readable file. Use when a package manager does not define scopes.
DEPENDENCY_MANIFEST_OF	Is to be used when SPDXRef-A is a manifest file that lists a set of dependencies for SPDXRef-B.	A file `package.json` is the dependency manifest of a package `foo`. Note that only one manifest should be used to define the same dependency graph.
BUILD_DEPENDENCY_OF	Is to be used when SPDXRef-A is a build dependency of SPDXRef-B.	A is in the `compile` scope of B in a Maven project.
DEV_DEPENDENCY_OF	Is to be used when SPDXRef-A is a development dependency of SPDXRef-B.	A is in the `devDependencies` scope of B in a Maven project.
OPTIONAL_DEPENDENCY_OF	Is to be used when SPDXRef-A is an optional dependency of SPDXRef-B.	Use when building the code will proceed even if a dependency cannot be found, fails to install, or is only installed on a specific platform. For example, A is in the `optionalDependencies` scope of npm project B.
PROVIDED_DEPENDENCY_OF	Is to be used when SPDXRef-A is a to be provided dependency of SPDXRef-B.	A is in the `provided` scope of B in a Maven project, indicating that the project expects it to be provided, for instance, by the container or JDK.
TEST_DEPENDENCY_OF	Is to be used when SPDXRef-A is a test dependency of SPDXRef-B.	A is in the `test` scope of B in a Maven project.

Relationship	Description	Example
RUNTIME_DEPENDENCY_OF	Is to be used when SPDXRef-A is a dependency required for the execution of SPDXRef-B.	A is in the `runtime` scope of B in a Maven project.
EXAMPLE_OF	Is to be used when SPDXRef-A is an example of SPDXRef-B.	The file or snippet that illustrates how to use an application or library.
GENERATES	Is to be used when SPDXRef-A generates SPDXRef-B.	A SOURCE file `makefile.mk` generates a BINARY file `a.out`
GENERATED_FROM	Is to be used when SPDXRef-A was generated from SPDXRef-B.	A BINARY file `a.out` has been generated from a SOURCE file `makefile.mk`. A BINARY file `foolib.a` is generated from a SOURCE file `bar.c`.
ANCESTOR_OF	Is to be used when SPDXRef-A is an ancestor (same lineage but pre-dates) SPDXRef-B.	A SOURCE file `makefile.mk` is a version of the original ancestor SOURCE file ‘makefile2.mk’
DESCENDANT_OF	Is to be used when SPDXRef-A is a descendant of (same lineage but postdates) SPDXRef-B.	A SOURCE file `makefile2.mk` is a descendant of the original SOURCE file ‘makefile.mk’
VARIANT_OF	Is to be used when SPDXRef-A is a variant of (same lineage but not clear which came first) SPDXRef-B.	A SOURCE file `makefile2.mk` is a variant of SOURCE file `makefile.mk` if they differ by some edit, but there is no way to tell which came first (no reliable date information).
DISTRIBUTION_ARTIFACT	Is to be used when distributing SPDXRef-A requires that SPDXRef-B also be distributed.	A BINARY file `foo.o` requires that the ARCHIVE file `bar-sources.tgz` be made available on distribution.
PATCH_FOR	Is to be used when SPDXRef-A is a patch file for (to be applied to) SPDXRef-B.	A SOURCE file `foo.diff` is a patch file for SOURCE file `foo.c`.
PATCH_APPLIED	Is to be used when SPDXRef-A is a patch file that has been applied to SPDXRef-B.	A SOURCE file `foo.diff` is a patch file that has been applied to SOURCE file ‘foo-patched.c’.

Relationship	Description	Example
COPY_OF	Is to be used when SPDXRef-A is an exact copy of SPDXRef-B.	A BINARY file `alib.a` is an exact copy of BINARY file `a2lib.a`.
FILE_ADDED	Is to be used when SPDXRef-A is a file that was added to SPDXRef-B.	A SOURCE file `foo.c` has been added to package ARCHIVE `bar.tgz`.
FILE_DELETED	Is to be used when SPDXRef-A is a file that was deleted from SPDXRef-B.	A SOURCE file `foo.diff` has been deleted from package ARCHIVE `bar.tgz`.
FILE_MODIFIED	Is to be used when SPDXRef-A is a file that was modified from SPDXRef-B.	A SOURCE file `foo.c` has been modified from SOURCE file `foo.orig.c`.
EXPANDED_FROM_ARCHIVE	Is to be used when SPDXRef-A is expanded from the archive SPDXRef-B.	A SOURCE file `foo.c`, has been expanded from the archive ARCHIVE file `xyz.tgz`.
DYNAMIC_LINK	Is to be used when SPDXRef-A dynamically links to SPDXRef-B.	An APPLICATION file 'myapp' dynamically links to BINARY file `zlib.so`.
STATIC_LINK	Is to be used when SPDXRef-A statically links to SPDXRef-B.	An APPLICATION file 'myapp' statically links to BINARY `zlib.a`.
DATA_FILE_OF	Is to be used when SPDXRef-A is a data file used in SPDXRef-B.	An IMAGE file 'kitty.jpg' is a data file of an APPLICATION 'hellokitty'.
TEST_CASE_OF	Is to be used when SPDXRef-A is a test case used in testing SPDXRef-B.	A SOURCE file `testMyCode.java` is a unit test file used to test an APPLICATION MyPackage.
BUILD_TOOL_OF	Is to be used when SPDXRef-A is used to build SPDXRef-B.	A SOURCE file `makefile.mk` is used to build an APPLICATION 'zlib'.
DEV_TOOL_OF	Is to be used when SPDXRef-A is used as a development tool for SPDXRef-B.	Any tool used for development such as a code debugger.

Relationship	Description	Example
TEST_OF	Is to be used when SPDXRef-A is used for testing SPDXRef-B.	Generic relationship for cases where it's clear that something is used for testing but unclear whether it's TEST_CASE_OF or TEST_TOOL_OF.
TEST_TOOL_OF	Is to be used when SPDXRef-A is used as a test tool for SPDXRef-B.	Any tool used to test the code such as ESlint.
DOCUMENTATION_OF	Is to be used when SPDXRef-A provides documentation of SPDXRef-B.	A DOCUMENTATION file `readme.txt` documents the APPLICATION ‘zlib’.
OPTIONAL_COMPONENT_OF	Is to be used when SPDXRef-A is an optional component of SPDXRef-B.	A SOURCE file `fool.c` (which is in the contributors directory) may or may not be included in the build of APPLICATION ‘atthebar’.
METAFILE_OF	Is to be used when SPDXRef-A is a metafile of SPDXRef-B.	A SOURCE file `pom.xml` is a metafile of the APPLICATION ‘Apache Xerces’.
PACKAGE_OF	Is to be used when SPDXRef-A is used as a package as part of SPDXRef-B.	A Linux distribution contains an APPLICATION package gawk as part of the distribution MyLinuxDistro.
AMENDS	Is to be used when (current) SPDXRef-DOCUMENT amends the SPDX information in SPDXRef-B.	(Current) SPDX document A version 2 contains a correction to a previous version of the SPDX document A version 1. Note the reserved identifier SPDXRef-DOCUMENT for the current document is required.
PREREQUISITE_FOR	Is to be used when SPDXRef-A is a prerequisite for SPDXRef-B.	A library `bar.dll` is a prerequisite or dependency for APPLICATION `foo.exe`
HAS_PREREQUISITE	Is to be used when SPDXRef-A has as a prerequisite SPDXRef-B.	An APPLICATION `foo.exe` has prerequisite or dependency on `bar.dll`
OTHER	Is to be used for a relationship which has not been defined in the formal SPDX specification. A description of the	

Relationship	Description	Example
	relationship should be included in the Relationship comments field.	

The metadata for the relationship field is shown in Table 69.

Table 69 — Metadata for the relationship field

Attribute	Value
Required	No
Cardinality	0..* see `DESCRIBES` relationship for one mandatory case.
Format	["DocumentRef-"[idstring]":"]SPDXID <relationship> ["DocumentRef-"[idstring]":"]`SPDXID` | `NONE` | `NOASSERTION` where "DocumentRef-" `[idstring]` ":" is an optional reference to an external SPDX document as described in 6.6 where `SPDXID` is a string containing letters, numbers, `.` and/or −. as described in 6.3, 7.2, 8.2. where `<relationship>` is one of the documented relationship types in Table 68. where `NONE` can be used to explicitly indicate there are NO other relationships. where `NOASSERTION` can be used to explicitly indicate it is not clear if there are relationships that may apply or not.

11.1.2 Intent

Here, this field is a reasonable estimation of the relation between two identified elements (i.e. files or packages, or documents), from a developer perspective.

11.1.3 Examples

EXAMPLE 1 Tag: `Relationship:`

```
Relationship: SPDXRef-grep CONTAINS SPDXRef-make

RelationshipComment: Package grep contains file make

Relationship: SPDXRef-DOCUMENT AMENDS DocumentRef-SPDXA:SPDXRef-DOCUMENT

RelationshipComment: This current document is an amendment of the SPDXA d
ocument.

Relationship: SPDXRef-CarolCompression DEPENDS_ON NONE

RelationshipComment: The package CarolCompression can be considered as a
root with no dependencies.
```

```
Relationship: SPDXRef-BobBrowser CONTAINS NOASSERTION

RelationshipComment: The package BobBrowser may have other packages embed
ded in it, but the author has insufficient information to treat this as o
ther than unknown at this point in time.
```

EXAMPLE 2 RDF: Property `spdx:relationship` in any `spdx:SpdxDocument`, `spdx:Package`, `spdx:File` or `spdx:Snippet`

```
<File rdf:about="#SPDXRef-45">
  <relationship>
    <Relationship>
      <relatedSpdxElement>
        <File rdf:about="http://spdx.org/spdxdocs/spdx-tools-v1.2-3F2504E
0-4F89-41D3-9A0C-0305E82..."/>
      </relatedSpdxElement>
      <relationshipType>http://spdx.org/rdf/terms#relationshipType_contai
ns
      </relationshipType>
    </Relationship>
  </relationship>
  ...
</File>
```

11.2 Relationship comment field

11.2.1 Description

This field provides a place for the SPDX document creator to record any general comments about the relationship. The metadata for the relationship comment field is shown in Table 70.

Table 70 — Metadata for the relationship comment field

Attribute	Value
Required	No
Cardinality	0..1
Format	Free form text that may span multiple lines, refers only to the immediately preceding relationship.

11.2.2 Intent

Here, the intent is to provide the recipient of the SPDX document with more information determined after careful analysis of the relationship between two elements in an SPDX document.

11.2.3 Examples

EXAMPLE 1 Tag: `RelationshipComment:`

In `tag:value` format multiple lines are delimited by `<text> .. </text>`.

A `RelationshipComment:` shall be the line immediately after a “Relationship:”

```
RelationshipComment: <text>The package foo.tgz is a pre-requisite for bui
lding executable bar.</text>
```

EXAMPLE 2 RDF: Property `rdfs:comment` in class `spdx:Relationship`

```
<Relationship rdf:about="...">
  <rdfs:comment>
    The package foo.tgz is a pre-requisite for building executable bar.
  </rdfs:comment>

  ...

</Relationship>
```

12 Annotations information section

12.1 Annotator field

12.1.1 Description

This field identifies the person, organization or tool that has commented on a snippet, file, package, or the entire document. The metadata for the annotator field is shown in Table 71.

Table 71 — Metadata for the annotator field

Attribute	Value
Required	Conditional
Cardinality	0..1 conditional (Mandatory, one), if there is an Annotation.

Format	Single line of text with the following keywords.
	`"Person: person name" and optional "(email)"`
	`"Organization: organization" and optional "(email)"`
	`"Tool: tool identifier - version"`

12.1.2 Intent

It may also be important for participants in the software supply chain to validate and add information on ambiguous snippets, files and packages.

12.1.3 Examples

EXAMPLE 1 Tag: `Annotator:`

```
Annotator: Person: Jane Doe ()
```

EXAMPLE 2 RDF: Property `spdx:annotator` in class `spdx:Annotation`

```
<Annotation>
    <annotator> Person: Jane Doe () </annotator>
</Annotation>
```

12.2 Annotation date field

12.2.1 Description

Identify when the comment was made. This shall be specified according to the combined date and time in the UTC format, as specified in the ISO 8601 standard. The metadata for the annotation date field is shown in Table 72.

Table 72 — Metadata for the annotation date field

Attribute	Value
Required	Conditional
Cardinality	0..1 conditional (Mandatory, one), if there is an Annotation.
Format	`YYYY-MM-DDThh:mm:ssZ` where: — `YYYY` is year — `MM` is month with leading zero — `DD` is day with leading zero — `T` is delimiter for time — `hh` is hours with leading zero in 24-hour time — `mm` is minutes with leading zero — `ss` is seconds with leading zero — `Z` is universal time indicator

12.2.2 Intent

Here, the Annotation Date can serve as a verification as to when the actual review was done.

12.2.3 Examples

EXAMPLE 1 Tag: `AnnotationDate:`

```
AnnotationDate: 2010-01-29T18:30:22Z
```

EXAMPLE 2 RDF: Property `spdx:annotationDate` in class `spdx:Annotation`

```
</Annotation>
    <annotationDate> 2010-01-29T18:30:22Z </annotation Date>
</Annotation>
```

12.3 Annotation type field

12.3.1 Description

This field describes the type of annotation. Annotations are usually created when someone reviews the SPDX document, and if this is the case the annotation type should be `REVIEW`. If the author wants to store extra information about one of the elements during creation, it is recommended to use the type of `OTHER`. The metadata for the annotation type field is shown in Table 73.

Table 73 — Metadata for the annotation type field

Attribute	Value
Required	Conditional
Cardinality	0..1 conditional (Mandatory, one), if there is an Annotation.
Format	`REVIEW` | `OTHER`

12.3.2 Intent

This allows the type of annotation to be recorded.

12.3.3 Examples

EXAMPLE 1 Tag: `AnnotationType:`

```
AnnotationType: REVIEW
```

EXAMPLE 2 RDF: Property `spdx:annotationType` in class `spdx:Annotation`

```
<Annotation>
    <annotationType rdf:resource
      ="http://spdx.org/rdf/terms#annotationType_other"/>
</Annotation>
```

12.4 SPDX identifier reference field

12.4.1 Description

Uniquely identify the element in an SPDX document which is being referenced. These may be referenced internally and externally with the addition of the SPDX document identifier. The metadata for the SPDX identifier reference field is shown in Table 74.

Table 74 — Metadata for the SPDX identifier reference field

Attribute	Value
Required	Conditional
Cardinality	0..1 conditional (Mandatory, one), if there is an Annotation.
Format	`[DocumentRef-[idstring]:]SPDXID` where: `["DocumentRef-"[idstring]":"]` is an optional reference to an external SPDX document as described in 6.6 `SPDXID` is a unique string containing letters, numbers, `.` and/or − as described in 6.3, 7.2 and 8.2.

12.4.2 Intent

There may be several versions of the same snippet, package or file within an SPDX document. Each element needs to be able to be referred to uniquely so that relationships between elements can be clearly articulated.

12.4.3 Examples

EXAMPLE 1 Tag: `SPDXREF:`

```
SPDXREF: SPDXRef-45
```

```
SPDXREF: DocumentRef-spdx-tool-1.2:SPDXRef-5
```

EXAMPLE 2 RDF:

For RDF, the annotations are a property of the SPDX document, package, file, or snippet they are annotating.

```
<File rdf:about="#SPDXRef-45">
    <annotation>
        <Annotation>
            ...
        </Annotation>
    </annotation>
</File>
```

12.5 Annotation comment field

12.5.1 Description

This required free form text field permits the annotator to provide commentary on the analysis. The metadata for the annotation comment field is shown in Table 75.

Table 75 — Metadata for the annotation comment field

Attribute	Value
Required	Conditional
Cardinality	0..1 conditional (Mandatory, one), if there is an Annotation.
Format	Free form text that may span multiple lines.

12.5.2 Intent

This allows the annotator to provide independent assessment and note any points where there is disagreement with the analysis.

12.5.3 Examples

EXAMPLE 1 Tag: `AnnotationComment:`

In `tag:value` format multiple lines are delimited by `<text> .. </text>`.

```
AnnotationComment: <text>All of the licenses seen in the file, are matchi
ng what was seen during manual inspection. There are some terms that can
influence the concluded license, and some alternatives may be possible, b
ut the concluded license is one of the options.</text>
```

EXAMPLE 2 RDF: Property `rdfs:comment` in class `spdx:Annotation`

```
<Annotation>
    <rdfs:comment>All of the licenses seen in the file, are matching what
was seen during manual inspection. There are some terms that can influenc
e the concluded license, and some alternatives may be possible, but the c
oncluded license is one of the options. </rdfs:comment>
</Annotation>
```

13 Review information section (deprecated)

13.1 Reviewer field (deprecated)

This field has been deprecated since SPDX 2.0.

13.1.1 Description

This field identifies the person, organization or tool that has reviewed the SPDX document. This field is optional and thus there is no requirement for any reviewer to add a set of review information to the SPDX document. This can be considered as an equivalent to “signed off” or “reviewed by.” Additional reviewers can be added after the original version of the SPDX document is created and be appended to the original SPDX document. The metadata for the reviewer field is shown in Table 76.

Table 76 — Metadata for the reviewer field

Attribute	Value
Required	No
Cardinality	0..1

Format	Single line of text with the following keywords.
	`"Person: person name" and optional "(email)"`
	`"Organization: organization" and optional "(email)"`
	`"Tool: tool identifier - version"`

13.1.2 Intent

Here, as time progresses certain reviewers will begin to gain credibility as reliable. This field intends to make such information transparent. It may also be important for participants in the software supply chain to validate whether upstream providers have reviewed the SPDX document.

13.1.3 Examples

EXAMPLE 1 Tag: `Reviewer:`

```
Reviewer: Person: Jane Doe ()
```

EXAMPLE 2 RDF: Property `spdx:reviewer` in class `spdx:Review`

```
<Review>
    <reviewer> Person: Jane Doe () </reviewer>
</Review>
```

13.2 Review date field (deprecated)

This field has been deprecated since SPDX 2.0.

13.2.1 Description

Identify when the review was done. This shall be specified according to the combined date and time in the UTC format, as specified in the ISO 8601 standard. The metadata for the review date field is shown in Table 77.

Table 77 — Metadata for the review date field

Attribute	Value
Required	Conditional
Cardinality	0..1 conditional (Mandatory, one), if there is a Reviewer.
Format	`YYYY-MM-DDThh:mm:ssZ` where: — `YYYY` is year — `MM` is month with leading zero — `DD` is day with leading zero — `T` is delimiter for time — `hh` is hours with leading zero in 24-hour time — `mm` is minutes with leading zero — `ss` is seconds with leading zero — `Z` is universal time indicator

13.2.2 Intent

Here, the `ReviewDate` can serve as a verification as to when the actual review was done.

13.2.3 Examples

EXAMPLE 1 Tag: `ReviewDate:`

```
ReviewDate: 2010-01-29T18:30:22Z
```

EXAMPLE 2 RDF: Property `spdx:reviewDate` in class `spdx:Review`

```
<Review>
    <reviewDate> 2010-01-29T18:30:22Z </reviewDate>
</Review>
```

13.3 Review comment field (deprecated)

This field is deprecated since SPDX 2.0.

13.3.1 Description

This optional free form text field permits the reviewer to provide commentary on the analysis. The metadata for the review comment field is shown in Table 78.

Table 78 — Metadata for the review comment field

Attribute	Value
Required	No
Cardinality	0..1
Format	Free form text that may span multiple lines.

13.3.2 Intent

This allows the reviewer to provide independent assessment and note any points where there is disagreement with the analysis.

13.3.3 Examples

EXAMPLE 1 Tag: `ReviewComment:`

In `tag:value` format multiple lines are delimited by `<text> .. </text>`.

```
ReviewComment: <text>All of the licenses seen in the file, are matching w
hat was seen during manual inspection. There are some terms that can infl
uence the concluded license, and some alternatives may be possible, but t
he concluded license is one of the options.</text>
```

EXAMPLE 2 RDF: Property `rdfs:comment` in class `spdx:Review`

```
<Review>
    <rdfs:comment>All of the licenses seen in the file, are matching what
was seen during manual inspection. There are some terms that can influenc
e the concluded license, and some alternatives may be possible, but the c
oncluded license is one of the options.</rdfs:comment>
</Review>
```

Annex A
(Informative)

SPDX license list

The SPDX License List is a list of commonly found licenses and exceptions used for open source and other collaborative software. The purpose of the SPDX License List is to enable easy and efficient identification of such licenses and exceptions in an SPDX document (or elsewhere). The SPDX License List includes a standardized short identifier, full name for each license, vetted license text, other basic information, and a canonical permanent URL for each license and exception. By providing a short identifier, users can efficiently refer to a license without having to redundantly reproduce the full license. License exceptions can be used with the License Expression Syntax operator, "WITH" to create a license with an exception.

— License Exceptions: The list of commonly found exceptions to open source licenses, which can be used with the License Expression operator, "WITH" to create a license with an exception.

— License List Repository: The HTML pages you see here are generated from the SPDX license-list-XML Repository. The repository contains XML files for all licenses, deprecated licenses, and license exceptions.

— Overview: For general information about the SPDX License List, including principles for inclusion of a license and an explanation of the fields contained on the list.

— Matching Guidelines: Guidelines for what constitutes a license match to the SPDX License List. For licenses that include markup, the license text on the HTML pages here will display omitable text in blue and replaceable text in red (see Guideline #2 for more information).

— Request New License: For instructions on how to propose additional licenses or license exceptions be added to the SPDX License List.

The following table contains the full names and short identifiers for the SPDX License List, v3.8 which was released 2020-02-09. For the full and most up-to-date version of the SPDX License List as well as other related information, please see http://spdx.org/licenses/

A.1 Licenses with short identifiers

The OSI column in Table A.1 refers to Open Source Initiative licenses [see Reference 2].

Table A.1 — License names and corresponding short identifiers

Full name of license	Short identifier	OSI?
BSD Zero Clause License	0BSD	Y
Attribution Assurance License	AAL	Y
Abstyles License	Abstyles	
Adobe Systems Incorporated Source Code License Agreement	Adobe-2006	

Full name of license	Short identifier	OSI?
Adobe Glyph List License	Adobe-Glyph	
Amazon Digital Services License	ADSL	
Academic Free License v1.1	AFL-1.1	Y
Academic Free License v1.2	AFL-1.2	Y
Academic Free License v2.0	AFL-2.0	Y
Academic Free License v2.1	AFL-2.1	Y
Academic Free License v3.0	AFL-3.0	Y
Afmparse License	Afmparse	
Affero General Public License v1.0 only	AGPL-1.0-only	
Affero General Public License v1.0 or later	AGPL-1.0-or-later	
GNU Affero General Public License v3.0 only	AGPL-3.0-only	Y
GNU Affero General Public License v3.0 or later	AGPL-3.0-or-later	Y
Aladdin Free Public License	Aladdin	
AMD's plpa_map.c License	AMDPLPA	
Apple MIT License	AML	
Academy of Motion Picture Arts and Sciences BSD	AMPAS	
ANTLR Software Rights Notice	ANTLR-PD	
Apache License 1.0	Apache-1.0	
Apache License 1.1	Apache-1.1	Y
Apache License 2.0	Apache-2.0	Y
Adobe Postscript AFM License	APAFML	
Adaptive Public License 1.0	APL-1.0	Y
Apple Public Source License 1.0	APSL-1.0	Y
Apple Public Source License 1.1	APSL-1.1	Y
Apple Public Source License 1.2	APSL-1.2	Y
Apple Public Source License 2.0	APSL-2.0	Y

Full name of license	Short identifier	OSI?
Artistic License 1.0	Artistic-1.0	Y
Artistic License 1.0 w/clause 8	Artistic-1.0-cl8	Y
Artistic License 1.0 (Perl)	Artistic-1.0-Perl	Y
Artistic License 2.0	Artistic-2.0	Y
Bahyph License	Bahyph	
Barr License	Barr	
Beerware License	Beerware	
BitTorrent Open Source License v1.0	BitTorrent-1.0	
BitTorrent Open Source License v1.1	BitTorrent-1.1	
SQLite Blessing	blessing	
Blue Oak Model License 1.0.0	BlueOak-1.0.0	
Borceux license	Borceux	
BSD 1-Clause License	BSD-1-Clause	
BSD 2-Clause "Simplified" License	BSD-2-Clause	Y
BSD 2-Clause FreeBSD License	BSD-2-Clause-FreeBSD	
BSD 2-Clause NetBSD License	BSD-2-Clause-NetBSD	
BSD-2-Clause Plus Patent License	BSD-2-Clause-Patent	Y
BSD 3-Clause "New" or "Revised" License	BSD-3-Clause	Y
BSD with attribution	BSD-3-Clause-Attribution	
BSD 3-Clause Clear License	BSD-3-Clause-Clear	
Lawrence Berkeley National Labs BSD variant license	BSD-3-Clause-LBNL	Y
BSD 3-Clause No Nuclear License	BSD-3-Clause-No-Nuclear-License	
BSD 3-Clause No Nuclear License 2014	BSD-3-Clause-No-Nuclear-License-2014	
BSD 3-Clause No Nuclear Warranty	BSD-3-Clause-No-Nuclear-Warranty	
BSD 3-Clause Open MPI variant	BSD-3-Clause-Open-MPI	

Full name of license	Short identifier	OSI?
BSD 4-Clause "Original" or "Old" License	BSD-4-Clause	
BSD-4-Clause (University of California-Specific)	BSD-4-Clause-UC	
BSD Protection License	BSD-Protection	
BSD Source Code Attribution	BSD-Source-Code	
Boost Software License 1.0	BSL-1.0	Y
bzip2 and libbzip2 License v1.0.5	bzip2-1.0.5	
bzip2 and libbzip2 License v1.0.6	bzip2-1.0.6	
Caldera License	Caldera	
Computer Associates Trusted Open Source License 1.1	CATOSL-1.1	Y
Creative Commons Attribution 1.0 Generic	CC-BY-1.0	
Creative Commons Attribution 2.0 Generic	CC-BY-2.0	
Creative Commons Attribution 2.5 Generic	CC-BY-2.5	
Creative Commons Attribution 3.0 Unported	CC-BY-3.0	
Creative Commons Attribution 4.0 International	CC-BY-4.0	
Creative Commons Attribution Non Commercial 1.0 Generic	CC-BY-NC-1.0	
Creative Commons Attribution Non Commercial 2.0 Generic	CC-BY-NC-2.0	
Creative Commons Attribution Non Commercial 2.5 Generic	CC-BY-NC-2.5	
Creative Commons Attribution Non Commercial 3.0 Unported	CC-BY-NC-3.0	
Creative Commons Attribution Non Commercial 4.0 International	CC-BY-NC-4.0	
Creative Commons Attribution Non Commercial No Derivatives 1.0 Generic	CC-BY-NC-ND-1.0	
Creative Commons Attribution Non Commercial No Derivatives 2.0 Generic	CC-BY-NC-ND-2.0	
Creative Commons Attribution Non Commercial No Derivatives 2.5 Generic	CC-BY-NC-ND-2.5	

Full name of license	Short identifier	OSI?
Creative Commons Attribution Non Commercial No Derivatives 3.0 Unported	CC-BY-NC-ND-3.0	
Creative Commons Attribution Non Commercial No Derivatives 4.0 International	CC-BY-NC-ND-4.0	
Creative Commons Attribution Non Commercial Share Alike 1.0 Generic	CC-BY-NC-SA-1.0	
Creative Commons Attribution Non Commercial Share Alike 2.0 Generic	CC-BY-NC-SA-2.0	
Creative Commons Attribution Non Commercial Share Alike 2.5 Generic	CC-BY-NC-SA-2.5	
Creative Commons Attribution Non Commercial Share Alike 3.0 Unported	CC-BY-NC-SA-3.0	
Creative Commons Attribution Non Commercial Share Alike 4.0 International	CC-BY-NC-SA-4.0	
Creative Commons Attribution No Derivatives 1.0 Generic	CC-BY-ND-1.0	
Creative Commons Attribution No Derivatives 2.0 Generic	CC-BY-ND-2.0	
Creative Commons Attribution No Derivatives 2.5 Generic	CC-BY-ND-2.5	
Creative Commons Attribution No Derivatives 3.0 Unported	CC-BY-ND-3.0	
Creative Commons Attribution No Derivatives 4.0 International	CC-BY-ND-4.0	
Creative Commons Attribution Share Alike 1.0 Generic	CC-BY-SA-1.0	
Creative Commons Attribution Share Alike 2.0 Generic	CC-BY-SA-2.0	
Creative Commons Attribution Share Alike 2.5 Generic	CC-BY-SA-2.5	
Creative Commons Attribution Share Alike 3.0 Unported	CC-BY-SA-3.0	
Creative Commons Attribution Share Alike 4.0 International	CC-BY-SA-4.0	
Creative Commons Public Domain Dedication and Certification	CC-PDDC	
Creative Commons Zero v1.0 Universal	CC0-1.0	
Common Development and Distribution License 1.0	CDDL-1.0	Y
Common Development and Distribution License 1.1	CDDL-1.1	

Full name of license	Short identifier	OSI?
Community Data License Agreement Permissive 1.0	CDLA-Permissive-1.0	
Community Data License Agreement Sharing 1.0	CDLA-Sharing-1.0	
CeCILL Free Software License Agreement v1.0	CECILL-1.0	
CeCILL Free Software License Agreement v1.1	CECILL-1.1	
CeCILL Free Software License Agreement v2.0	CECILL-2.0	
CeCILL Free Software License Agreement v2.1	CECILL-2.1	Y
CeCILL-B Free Software License Agreement	CECILL-B	
CeCILL-C Free Software License Agreement	CECILL-C	
CERN Open Hardware Licence v1.1	CERN-OHL-1.1	
CERN Open Hardware Licence v1.2	CERN-OHL-1.2	
Clarified Artistic License	ClArtistic	
CNRI Jython License	CNRI-Jython	
CNRI Python License	CNRI-Python	Y
CNRI Python Open Source GPL Compatible License Agreement	CNRI-Python-GPL-Compatible	
Condor Public License v1.1	Condor-1.1	
copyleft-next 0.3.0	copyleft-next-0.3.0	
copyleft-next 0.3.1	copyleft-next-0.3.1	
Common Public Attribution License 1.0	CPAL-1.0	Y
Common Public License 1.0	CPL-1.0	Y
Code Project Open License 1.02	CPOL-1.02	
Crossword License	Crossword	
CrystalStacker License	CrystalStacker	
CUA Office Public License v1.0	CUA-OPL-1.0	Y
Cube License	Cube	
curl License	curl	
Deutsche Freie Software Lizenz	D-FSL-1.0	

Full name of license	Short identifier	OSI?
diffmark license	diffmark	
DOC License	DOC	
Dotseqn License	Dotseqn	
DSDP License	DSDP	
dvipdfm License	dvipdfm	
Educational Community License v1.0	ECL-1.0	Y
Educational Community License v2.0	ECL-2.0	Y
Eiffel Forum License v1.0	EFL-1.0	Y
Eiffel Forum License v2.0	EFL-2.0	Y
eGenix.com Public License 1.1.0	eGenix	
Entessa Public License v1.0	Entessa	Y
Eclipse Public License 1.0	EPL-1.0	Y
Eclipse Public License 2.0	EPL-2.0	Y
Erlang Public License v1.1	ErlPL-1.1	
Etalab Open License 2.0	etalab-2.0	
EU DataGrid Software License	EUDatagrid	Y
European Union Public License 1.0	EUPL-1.0	
European Union Public License 1.1	EUPL-1.1	Y
European Union Public License 1.2	EUPL-1.2	Y
Eurosym License	Eurosym	
Fair License	Fair	Y
Frameworx Open License 1.0	Frameworx-1.0	Y
FreeImage Public License v1.0	FreeImage	
FSF All Permissive License	FSFAP	
FSF Unlimited License	FSFUL	
FSF Unlimited License (with License Retention)	FSFULLR	

Full name of license	Short identifier	OSI?
Freetype Project License	FTL	
GNU Free Documentation License v1.1 only	GFDL-1.1-only	
GNU Free Documentation License v1.1 or later	GFDL-1.1-or-later	
GNU Free Documentation License v1.2 only	GFDL-1.2-only	
GNU Free Documentation License v1.2 or later	GFDL-1.2-or-later	
GNU Free Documentation License v1.3 only	GFDL-1.3-only	
GNU Free Documentation License v1.3 or later	GFDL-1.3-or-later	
Giftware License	Giftware	
GL2PS License	GL2PS	
3dfx Glide License	Glide	
Glulxe License	Glulxe	
gnuplot License	gnuplot	
GNU General Public License v1.0 only	GPL-1.0-only	
GNU General Public License v1.0 or later	GPL-1.0-or-later	
GNU General Public License v2.0 only	GPL-2.0-only	Y
GNU General Public License v2.0 or later	GPL-2.0-or-later	Y
GNU General Public License v3.0 only	GPL-3.0-only	Y
GNU General Public License v3.0 or later	GPL-3.0-or-later	Y
gSOAP Public License v1.3b	gSOAP-1.3b	
Haskell Language Report License	HaskellReport	
Historical Permission Notice and Disclaimer	HPND	Y
Historical Permission Notice and Disclaimer - sell variant	HPND-sell-variant	
IBM PowerPC Initialization and Boot Software	IBM-pibs	
ICU License	ICU	
Independent JPEG Group License	IJG	
ImageMagick License	ImageMagick	

Full name of license	Short identifier	OSI?
iMatix Standard Function Library Agreement	iMatix	
Imlib2 License	Imlib2	
Info-ZIP License	Info-ZIP	
Intel Open Source License	Intel	Y
Intel ACPI Software License Agreement	Intel-ACPI	
Interbase Public License v1.0	Interbase-1.0	
IPA Font License	IPA	Y
IBM Public License v1.0	IPL-1.0	Y
ISC License	ISC	Y
JasPer License	JasPer-2.0	
Japan Network Information Center License	JPNIC	
JSON License	JSON	
Licence Art Libre 1.2	LAL-1.2	
Licence Art Libre 1.3	LAL-1.3	
Latex2e License	Latex2e	
Leptonica License	Leptonica	
GNU Library General Public License v2 only	LGPL-2.0-only	Y
GNU Library General Public License v2 or later	LGPL-2.0-or-later	Y
GNU Lesser General Public License v2.1 only	LGPL-2.1-only	Y
GNU Lesser General Public License v2.1 or later	LGPL-2.1-or-later	Y
GNU Lesser General Public License v3.0 only	LGPL-3.0-only	Y
GNU Lesser General Public License v3.0 or later	LGPL-3.0-or-later	Y
Lesser General Public License For Linguistic Resources	LGPLLR	
libpng License	Libpng	
PNG Reference Library version 2	libpng-2.0	
libselinux public domain notice	libselinux-1.0	

Full name of license	Short identifier	OSI?
libtiff License	libtiff	
Licence Libre du Québec – Permissive version 1.1	LiLiQ-P-1.1	Y
Licence Libre du Québec – Réciprocité version 1.1	LiLiQ-R-1.1	Y
Licence Libre du Québec – Réciprocité forte version 1.1	LiLiQ-Rplus-1.1	Y
Linux Kernel Variant of OpenIB.org license	Linux-OpenIB	
Lucent Public License Version 1.0	LPL-1.0	Y
Lucent Public License v1.02	LPL-1.02	Y
LaTeX Project Public License v1.0	LPPL-1.0	
LaTeX Project Public License v1.1	LPPL-1.1	
LaTeX Project Public License v1.2	LPPL-1.2	
LaTeX Project Public License v1.3a	LPPL-1.3a	
LaTeX Project Public License v1.3c	LPPL-1.3c	Y
MakeIndex License	MakeIndex	
The MirOS Licence	MirOS	Y
MIT License	MIT	Y
MIT No Attribution	MIT-0	
Enlightenment License (e16)	MIT-advertising	
CMU License	MIT-CMU	
enna License	MIT-enna	
feh License	MIT-feh	
MIT +no-false-attribs license	MITNFA	
Motosoto License	Motosoto	Y
mpich2 License	mpich2	
Mozilla Public License 1.0	MPL-1.0	Y
Mozilla Public License 1.1	MPL-1.1	Y
Mozilla Public License 2.0	MPL-2.0	Y

Full name of license	Short identifier	OSI?
Mozilla Public License 2.0 (no copyleft exception)	MPL-2.0-no-copyleft-exception	Y
Microsoft Public License	MS-PL	Y
Microsoft Reciprocal License	MS-RL	Y
Matrix Template Library License	MTLL	
Mulan Permissive Software License, Version 1	MulanPSL-1.0	
Multics License	Multics	Y
Mup License	Mup	
NASA Open Source Agreement 1.3	NASA-1.3	Y
Naumen Public License	Naumen	Y
Net Boolean Public License v1	NBPL-1.0	
University of Illinois/NCSA Open Source License	NCSA	Y
Net-SNMP License	Net-SNMP	
NetCDF license	NetCDF	
Newsletr License	Newsletr	
Nethack General Public License	NGPL	Y
Norwegian Licence for Open Government Data	NLOD-1.0	
No Limit Public License	NLPL	
Nokia Open Source License	Nokia	Y
Netizen Open Source License	NOSL	
Noweb License	Noweb	
Netscape Public License v1.0	NPL-1.0	
Netscape Public License v1.1	NPL-1.1	
Non-Profit Open Software License 3.0	NPOSL-3.0	Y
NRL License	NRL	
NTP License	NTP	Y
NTP No Attribution	NTP-0	

Full name of license	Short identifier	OSI?
Open CASCADE Technology Public License	OCCT-PL	
OCLC Research Public License 2.0	OCLC-2.0	Y
ODC Open Database License v1.0	ODbL-1.0	
Open Data Commons Attribution License v1.0	ODC-By-1.0	
SIL Open Font License 1.0	OFL-1.0	
SIL Open Font License 1.0 with no Reserved Font Name	OFL-1.0-no-RFN	
SIL Open Font License 1.0 with Reserved Font Name	OFL-1.0-RFN	
SIL Open Font License 1.1	OFL-1.1	Y
SIL Open Font License 1.1 with no Reserved Font Name	OFL-1.1-no-RFN	Y
SIL Open Font License 1.1 with Reserved Font Name	OFL-1.1-RFN	Y
Open Government Licence - Canada	OGL-Canada-2.0	
Open Government Licence v1.0	OGL-UK-1.0	
Open Government Licence v2.0	OGL-UK-2.0	
Open Government Licence v3.0	OGL-UK-3.0	
Open Group Test Suite License	OGTSL	Y
Open LDAP Public License v1.1	OLDAP-1.1	
Open LDAP Public License v1.2	OLDAP-1.2	
Open LDAP Public License v1.3	OLDAP-1.3	
Open LDAP Public License v1.4	OLDAP-1.4	
Open LDAP Public License v2.0 (or possibly 2.0A and 2.0B)	OLDAP-2.0	
Open LDAP Public License v2.0.1	OLDAP-2.0.1	
Open LDAP Public License v2.1	OLDAP-2.1	
Open LDAP Public License v2.2	OLDAP-2.2	
Open LDAP Public License v2.2.1	OLDAP-2.2.1	
Open LDAP Public License 2.2.2	OLDAP-2.2.2	
Open LDAP Public License v2.3	OLDAP-2.3	

Full name of license	Short identifier	OSI?
Open LDAP Public License v2.4	OLDAP-2.4	
Open LDAP Public License v2.5	OLDAP-2.5	
Open LDAP Public License v2.6	OLDAP-2.6	
Open LDAP Public License v2.7	OLDAP-2.7	
Open LDAP Public License v2.8	OLDAP-2.8	
Open Market License	OML	
OpenSSL License	OpenSSL	
Open Public License v1.0	OPL-1.0	
OSET Public License version 2.1	OSET-PL-2.1	Y
Open Software License 1.0	OSL-1.0	Y
Open Software License 1.1	OSL-1.1	
Open Software License 2.0	OSL-2.0	Y
Open Software License 2.1	OSL-2.1	Y
Open Software License 3.0	OSL-3.0	Y
The Parity Public License 6.0.0	Parity-6.0.0	
ODC Public Domain Dedication & License 1.0	PDDL-1.0	
PHP License v3.0	PHP-3.0	Y
PHP License v3.01	PHP-3.01	
Plexus Classworlds License	Plexus	
PostgreSQL License	PostgreSQL	Y
Python Software Foundation License 2.0	PSF-2.0	
psfrag License	psfrag	
psutils License	psutils	
Python License 2.0	Python-2.0	Y
Qhull License	Qhull	
Q Public License 1.0	QPL-1.0	Y

Full name of license	Short identifier	OSI?
Rdisc License	Rdisc	
Red Hat eCos Public License v1.1	RHeCos-1.1	
Reciprocal Public License 1.1	RPL-1.1	Y
Reciprocal Public License 1.5	RPL-1.5	Y
RealNetworks Public Source License v1.0	RPSL-1.0	Y
RSA Message-Digest License	RSA-MD	
Ricoh Source Code Public License	RSCPL	Y
Ruby License	Ruby	
Sax Public Domain Notice	SAX-PD	
Saxpath License	Saxpath	
SCEA Shared Source License	SCEA	
Sendmail License	Sendmail	
Sendmail License 8.23	Sendmail-8.23	
SGI Free Software License B v1.0	SGI-B-1.0	
SGI Free Software License B v1.1	SGI-B-1.1	
SGI Free Software License B v2.0	SGI-B-2.0	
Solderpad Hardware License v0.5	SHL-0.5	
Solderpad Hardware License, Version 0.51	SHL-0.51	
Simple Public License 2.0	SimPL-2.0	Y
Sun Industry Standards Source License v1.1	SISSL	Y
Sun Industry Standards Source License v1.2	SISSL-1.2	
Sleepycat License	Sleepycat	Y
Standard ML of New Jersey License	SMLNJ	
Secure Messaging Protocol Public License	SMPPL	
SNIA Public License 1.1	SNIA	
Spencer License 86	Spencer-86	

Full name of license	Short identifier	OSI?
Spencer License 94	Spencer-94	
Spencer License 99	Spencer-99	
Sun Public License v1.0	SPL-1.0	Y
SSH OpenSSH license	SSH-OpenSSH	
SSH short notice	SSH-short	
Server Side Public License, v 1	SSPL-1.0	
SugarCRM Public License v1.1.3	SugarCRM-1.1.3	
Scheme Widget Library (SWL) Software License Agreement	SWL	
TAPR Open Hardware License v1.0	TAPR-OHL-1.0	
TCL/TK License	TCL	
TCP Wrappers License	TCP-wrappers	
TMate Open Source License	TMate	
TORQUE v2.5+ Software License v1.1	TORQUE-1.1	
Trusster Open Source License	TOSL	
Technische Universitaet Berlin License 1.0	TU-Berlin-1.0	
Technische Universitaet Berlin License 2.0	TU-Berlin-2.0	
Upstream Compatibility License v1.0	UCL-1.0	Y
Unicode License Agreement - Data Files and Software (2015)	Unicode-DFS-2015	
Unicode License Agreement - Data Files and Software (2016)	Unicode-DFS-2016	
Unicode Terms of Use	Unicode-TOU	
The Unlicense	Unlicense	
Universal Permissive License v1.0	UPL-1.0	Y
Vim License	Vim	
VOSTROM Public License for Open Source	VOSTROM	
Vovida Software License v1.0	VSL-1.0	Y

Full name of license	Short identifier	OSI?
W3C Software Notice and License (2002-12-31)	W3C	Y
W3C Software Notice and License (1998-07-20)	W3C-19980720	
W3C Software Notice and Document License (2015-05-13)	W3C-20150513	
Sybase Open Watcom Public License 1.0	Watcom-1.0	Y
Wsuipa License	Wsuipa	
Do What The F*ck You Want To Public License	WTFPL	
X11 License	X11	
Xerox License	Xerox	
XFree86 License 1.1	XFree86-1.1	
xinetd License	xinetd	
X.Net License	Xnet	Y
XPP License	xpp	
XSkat License	XSkat	
Yahoo! Public License v1.0	YPL-1.0	
Yahoo! Public License v1.1	YPL-1.1	
Zed License	Zed	
Zend License v2.0	Zend-2.0	
Zimbra Public License v1.3	Zimbra-1.3	
Zimbra Public License v1.4	Zimbra-1.4	
zlib License	Zlib	Y
zlib/libpng License with Acknowledgement	zlib-acknowledgement	
Zope Public License 1.1	ZPL-1.1	
Zope Public License 2.0	ZPL-2.0	Y
Zope Public License 2.1	ZPL-2.1	

A.2 Exceptions list

Table A.2 — Exception names and license links

Full name of exception	SPDX license exception
389 Directory Server Exception	389-exception
Autoconf exception 2.0	Autoconf-exception-2.0
Autoconf exception 3.0	Autoconf-exception-3.0
Bison exception 2.2	Bison-exception-2.2
Bootloader Distribution Exception	Bootloader-exception
Classpath exception 2.0	Classpath-exception-2.0
CLISP exception 2.0	CLISP-exception-2.0
DigiRule FOSS License Exception	DigiRule-FOSS-exception
eCos exception 2.0	eCos-exception-2.0
Fawkes Runtime Exception	Fawkes-Runtime-exception
FLTK exception	FLTK-exception
Font exception 2.0	Font-exception-2.0
FreeRTOS Exception 2.0	freertos-exception-2.0
GCC Runtime Library exception 2.0	GCC-exception-2.0
GCC Runtime Library exception 3.1	GCC-exception-3.1
GNU JavaMail exception	gnu-javamail-exception
GPL-3.0 Linking Exception	GPL-3.0-linking-exception
GPL-3.0 Linking Exception (with Corresponding Source)	GPL-3.0-linking-source-exception
GPL Cooperation Commitment 1.0	GPL-CC-1.0
i2p GPL+Java Exception	i2p-gpl-java-exception
Libtool Exception	Libtool-exception
Linux Syscall Note	Linux-syscall-note
LLVM Exception	LLVM-exception
LZMA exception	LZMA-exception
Macros and Inline Functions Exception	mif-exception

Full name of exception	SPDX license exception
OCaml LGPL Linking Exception	OCaml-LGPL-linking-exception
Open CASCADE Exception 1.0	OCCT-exception-1.0
OpenJDK Assembly exception 1.0	OpenJDK-assembly-exception-1.0
OpenVPN OpenSSL Exception	openvpn-openssl-exception
PS/PDF font exception (2017-08-17)	PS-or-PDF-font-exception-20170817
Qt GPL exception 1.0	Qt-GPL-exception-1.0
Qt LGPL exception 1.1	Qt-LGPL-exception-1.1
Qwt exception 1.0	Qwt-exception-1.0
Swift Exception	Swift-exception
U-Boot exception 2.0	u-boot-exception-2.0
Universal FOSS Exception, Version 1.0	Universal-FOSS-exception-1.0
WxWindows Library Exception 3.1	WxWindows-exception-3.1

A.3 Deprecated licenses

Table A.3 — Deprecated license short identifiers

Full name of license	Deprecated SDPX short identifier
Affero General Public License v1.0	AGPL-1.0
GNU Affero General Public License v3.0	AGPL-3.0
eCos license version 2.0	eCos-2.0
GNU Free Documentation License v1.1	GFDL-1.1
GNU Free Documentation License v1.2	GFDL-1.2
GNU Free Documentation License v1.3	GFDL-1.3
GNU General Public License v1.0 only	GPL-1.0
GNU General Public License v1.0 or later	GPL-1.0+
GNU General Public License v2.0 only	GPL-2.0
GNU General Public License v2.0 or later	GPL-2.0+
GNU General Public License v2.0 w/Autoconf exception	GPL-2.0-with-autoconf-exception

Full name of license	Deprecated SDPX short identifier
GNU General Public License v2.0 w/Bison exception	GPL-2.0-with-bison-exception
GNU General Public License v2.0 w/Classpath exception	GPL-2.0-with-classpath-exception
GNU General Public License v2.0 w/Font exception	GPL-2.0-with-font-exception
GNU General Public License v2.0 w/GCC Runtime Library exception	GPL-2.0-with-GCC-exception
GNU General Public License v3.0 only	GPL-3.0
GNU General Public License v3.0 or later	GPL-3.0+
GNU General Public License v3.0 w/Autoconf exception	GPL-3.0-with-autoconf-exception
GNU General Public License v3.0 w/GCC Runtime Library exception	GPL-3.0-with-GCC-exception
GNU Library General Public License v2 only	LGPL-2.0
GNU Library General Public License v2 or later	LGPL-2.0+
GNU Lesser General Public License v2.1 only	LGPL-2.1
GNU Library General Public License v2.1 or later	LGPL-2.1+
GNU Lesser General Public License v3.0 only	LGPL-3.0
GNU Lesser General Public License v3.0 or later	LGPL-3.0+
Nunit License	Nunit
Standard ML of New Jersey License	StandardML-NJ
wxWindows Library License	wxWindows

Annex B
(Informative)

License matching guidelines and templates

B.1 SPDX license list matching guidelines

The SPDX License List Matching Guidelines provide guidelines to be used for the purposes of matching licenses and license exceptions against those included on the SPDX License List. There is no intent here to make a judgment or interpretation, but merely to ensure that when one SPDX user identifies a license as "BSD-3-Clause," for example, it is indeed the same license as what someone else identifies as "BSD-3-Clause" and the same license as what is listed on the SPDX License List. Examples of how to apply some of the matching guidelines to a license or exception are provided via templates. Templates are comprised of technical markup within the master license text file to provide further or specific guidance to SPDX document creators or tool makers. Not all licenses or exceptions will have templates.

B.2 How these guidelines are applied

B.2.1 Purpose

To ensure consistent results by different SPDX document creators when matching license information that will be included in the License Information in File field. SPDX document creators or tools may match on the license or exception text itself, the official license header, or the SPDX License List short identifier.

B.2.2 Guideline: official license headers

The same matching guidelines used for license and exception text apply to the official license headers. Where applicable, an official license header template file with markup is included with the SPDX License List. Official license headers are defined by the SPDX License List as specific text specified within the license itself to be put in the header of files. (see https://spdx.org/spdx-license-list/license-list-overview for more info).

B.3 Substantive text

B.3.1 Purpose

To ensure that when matching licenses and exceptions to the SPDX License List, there is an appropriate balance between matching against the substantive text and disregarding parts of the text that do not alter the substantive text. Further guidelines of what can be disregarded or considered replaceable for purposes of matching are listed below here and in the subsequent specific guidelines. A conservative approach is taken in regard to rules about disregarded or replaceable text.

B.3.2 Guideline: verbatim text

License and exception text should be the same verbatim text (except for the guidelines stated here). The text should be in the same order, e.g., differently ordered paragraphs would not be considered a match.

B.3.3 Guideline: no additional text

Matched text should only include that found in the vetted license or exception text. Where a license or exception found includes additional text or clauses, this should not be considered a match.

B.3.4 Guideline: replaceable text

Some licenses include text that refers to the specific copyright holder or author, yet the rest of the license is exactly the same as a generic version. The intent here is to avoid the inclusion of a specific name in one part of the license resulting in a non-match where the license is otherwise an exact match (e.g., the third clause and disclaimer in the BSD licenses, or the third, fourth, and fifth clauses of Apache-1.1). In these cases, there should be a positive license match.

Text that can be considered replaceable for matching purposes is indicated in the SPDX License List template with mark-up and in the corresponding HTML pages with colored text. The text indicated as such can be replaced with similar values (e.g., a different name or generic term; different date) and still be considered a positive match. This rule also applies to text-matching in official license headers (see Guideline #1).

B.3.5 Guideline: omittable text

Some licenses have text that can simply be ignored. The intent here is to avoid the inclusion of certain text that is superfluous or irrelevant in regards to the substantive license text resulting in a non-match where the license is otherwise an exact match (e.g., directions on how to apply the license or other similar non-substantive exhibits). In these cases, there should be a positive license match.

Text that can be considered omittable for matching purposes is indicated in the SPDX License List template with mark-up and in the corresponding HTML pages with colored text. The license should be considered a match if the text indicated is present and matches OR the text indicated is missing altogether.

B.4 Whitespace

B.4.1 Purpose

To avoid the possibility of a non-match due to different spacing of words, line breaks, or paragraphs.

B.4.2 Guideline

All whitespace should be treated as a single blank space. Templates do not include markup for this guideline.

B.5 Capitalization

B.5.1 Purpose

To avoid the possibility of a non-match due to lowercase or uppercase letters in otherwise the same words.

B.5.2 Guideline

All upper case and lowercase letters should be treated as lower case letters. Templates do not include markup for this guideline.

B.6 Punctuation

B.6.1 Purpose

Because punctuation can change the meaning of a sentence, punctuation needs to be included in the matching process. License template files do not include markup for this guideline.

B.6.2 Guideline: punctuation

Punctuation should be matched, unless otherwise stated in these guidelines.

B.6.3 Guideline: hyphens, dashes

Any hyphen, dash, en dash, em dash, or other variation should be considered equivalent.

B.6.4 Guideline: quotes

Any variation of quotations (single, double, curly, etc.) should be considered equivalent.

B.7 Code comment indicators

B.7.1 Purpose

To avoid the possibility of a non-match due to the existence or absence of code comment indicators placed within the license text, e.g. at the start of each line of text.

B.7.2 Guideline

Any kind of code comment indicator or prefix which occurs at the beginning of each line in a matchable section should be ignored for matching purposes. Templates do not include markup for this guideline.

B.8 Bullets and numbering

B.8.1 Purpose

To avoid the possibility of a non-match due to the otherwise same license using bullets instead of numbers, number instead of letter, or no bullets instead of bullet, etc., for a list of clauses.

B.8.2 Guideline

Where a line starts with a bullet, number, letter, or some form of a list item (determined where list item is followed by a space, then the text of the sentence), ignore the list item for matching purposes. Templates do not include markup for this guideline.

B.9 Varietal word spelling

B.9.1 Purpose

English uses different spelling for some words. By identifying the spelling variations for words found or likely to be found in licenses, we avoid the possibility of a non-match due to the same word being spelled differently. This list is not meant to be an exhaustive list of all spelling variations, but meant to capture the words most likely to be found in open source software licenses.

B.9.2 Guideline

The words in each line of the text file available at https://github.com/spdx/license-list-XML/blob/master/equivalentwords.txt are considered equivalent and interchangeable. Templates do not include markup for this guideline.

B.10 Copyright symbol

B.10.1 Purpose

By having a rule regarding the use of "©", "(c)", or "copyright", we avoid the possibility of a mismatch based on these variations.

B.10.2 Guideline

"©", "(c)", or "Copyright" should be considered equivalent and interchangeable. Templates do not include markup for this guideline.

B.11 Copyright notice

B.11.1 Purpose

To avoid a license mismatch merely because the copyright notice (usually found above the actual license or exception text) is different. The copyright notice is important information to be recorded elsewhere in the SPDX document, but for the purposes of matching a license to the SPDX License List, it should be ignored because it is not part of the substantive license text.

B.11.2 Guideline

Ignore copyright notices. A copyright notice consists of the following elements, for example: "2012 Copyright, John Doe. All rights reserved." or "(c) 2012 John Doe." Templates may or may not include markup for this guideline.

B.12 License name or title

B.12.1 Purpose

To avoid a license mismatch merely because the name or title of the license is different than how the license is usually referred to or different than the SPDX full name. This also avoids a mismatch if the title or name of the license is simply not included.

B.12.2 Guideline

Ignore the license name or title for matching purposes, so long as what ignored is the title only and there is no additional substantive text added here. Templates do not include markup for this guideline.

B.13 Extraneous text at the end of a license

B.13.1 Purpose

To avoid a license mismatch merely because extraneous text that appears at the end of the terms of a license is different or missing. This also avoids a mismatch if the extraneous text merely serves as a license notice example and includes a specific copyright holder's name.

B.13.2 Guideline

Ignore any text that occurs after the obvious end of the license and does not include substantive text of the license, for example: text that occurs after a statement such as, "END OF TERMS AND CONDITIONS," or an exhibit or appendix that includes an example or instructions on to how to apply the license to your code. Do not apply this guideline or ignore text that is comprised of additional license terms (e.g., permitted additional terms under GPL-3.0, section 7). Templates do not include markup for this guideline.

B.14 HTTP protocol

B.14.1 Purpose

To avoid a license mismatch due to a difference in a hyperlink protocol (e.g. http vs. https).

B.14.2 Guideline

HTTP:// and HTTPS:// should be considered equivalent. Templates may or may not include markup for this guideline.

B.15 SPDX license list

B.15.1 Template access

The master files for the SPDX License List includes a spreadsheet listing all the licenses, deprecated licenses, and license exceptions; and the text for each license in a .txt file. These files are available in a Git repository at https://github.com/spdx/license-list-XML. Text that can be considered replaceable or omitable for matching purposes is indicated in the .txt file with markup as per the description below.

RDFa Access: The template text for the license can be accessed using the RDF tag licenseTemplate on the web page containing the license.

B.15.2 Template format

A template is composed of text with zero or more rules embedded in it.

A rule is a variable section of a license wrapped between double angle brackets "<<>>" and is composed of 4 fields. Each field is separated with a semi-colon ";". Rules cannot be embedded within other rules. Rule fields begin with a case sensitive tag followed by an equal sign "=".

Rule fields:

- type: indicates whether the text is replaceable or omittable as per Matching Guideline #2 ("Substantive Text").
  - Indicated by `<<var; . . . >>` or...
  - Indicated by `<<beginOptional; . . .>>` and `<<endOptional>>` respectively.
  - This field is the first field and is required.
- name: name of the field in the template.
  - This field is unique within each license template.
  - This field is required.
- original: the original text of the rule.
  - This field is required for a rule type: `<<var; . . . >>`
- match: a POSIX extended regular expression (ERE).
  - This field is required for a rule type: `<<var; . . . >>`

The POSIX ERE in the match field has the following restrictions and extensions:

```
Semicolons are escaped with \;
```

```
POSIX Bracket Extensions are not allowed
```

EXAMPLE:

```
<<var;name=organizationClause3;original=the copyright holder;match=.+>>
```

Annex C
(Normative)

RDF object model and identifier syntax

C.1 Introduction

SPDX Vocabulary Specification

See: http://spdx.org/rdf/ontology/spdx-2-2-1

Version: 2.2.1

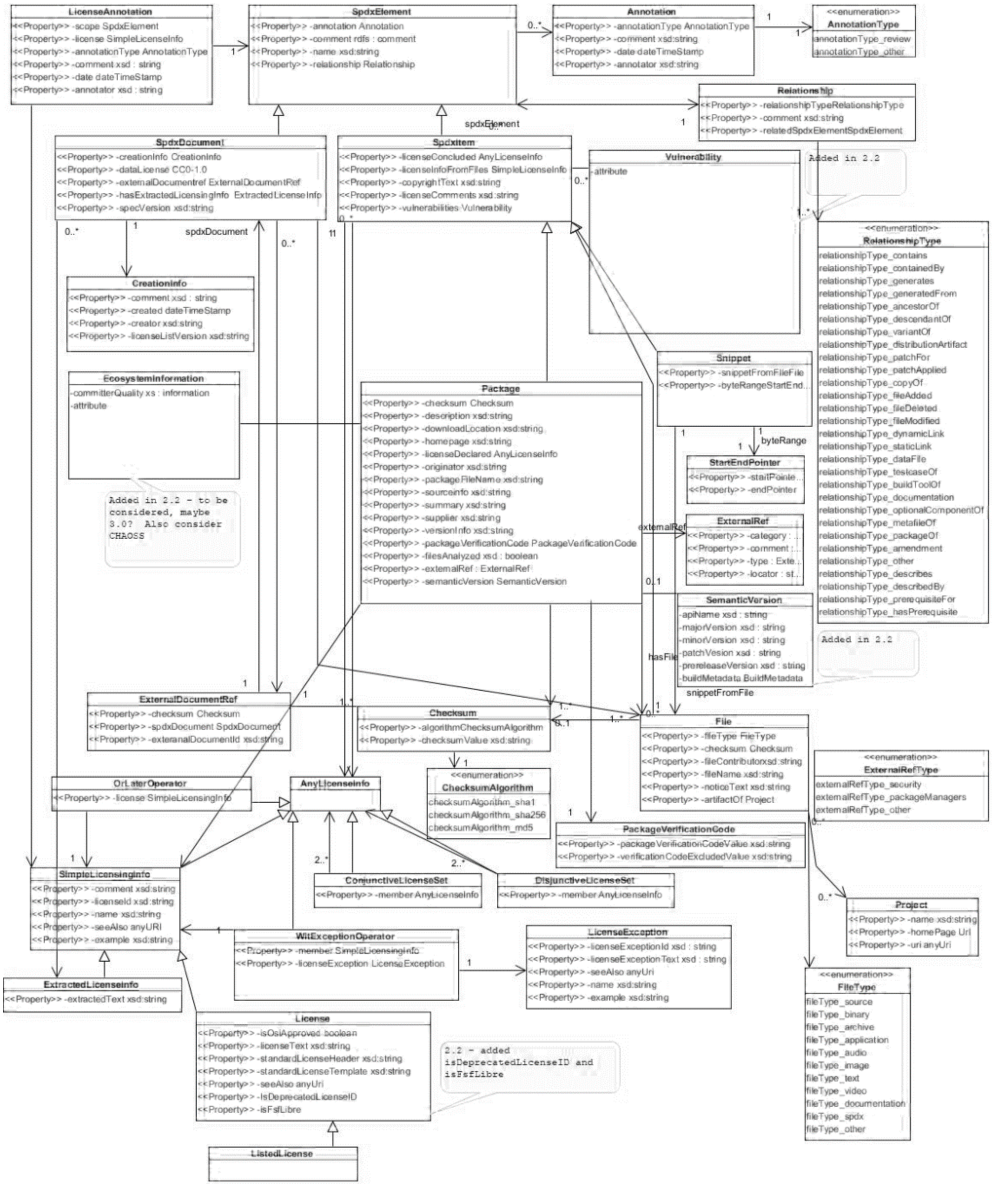
LicenseAnnotation
<<Property>> -scope SpdxElement
<<Property>> -license SimpleLicenseInfo
<<Property>> -annotationType AnnotationType
<<Property>> -comment xsd : string
<<Property>> -date dateTimeStamp
<<Property>> -annotator xsd : string
1
SpdxElement
<<Property>> -annotation Annotation
<<Property>> -comment rdfs : comment
<<Property>> -name xsd:string
<<Property>> -relationship Relationship
0..*
Annotation
<<Property>> -annotationType AnnotationType
<<Property>> -comment xsd:string
<<Property>> -date dateTimeStamp
<<Property>> -annotator xsd:string
1
1
<<enumeration>>
AnnotationType
annotationType_review
annotationType_other
Relationship
<<Property>> -relationshipTypeRelationshipType
<<Property>> -comment xsd:string
<<Property>> -relatedSpdxElementSpdxElement
1
spdxElement
0..*
SpdxDocument
<<Property>> -creationInfo CreationInfo
<<Property>> -dataLicense CC0-1.0
<<Property>> -externalDocumentref ExternalDocumentRef
<<Property>> -hasExtractedLicensingInfo ExtractedLicenseInfo
<<Property>> -specVersion xsd:string
0..*
1
spdxDocument
0..*
11
SpdxItem
<<Property>> -licenseConcluded AnyLicenseInfo
<<Property>> -licenseInfoFromFiles SimpleLicenseInfo
<<Property>> -copyrightText xsd:string
<<Property>> -licenseComments xsd:string
<<Property>> -vulnerabilities Vulnerability
0..*
0..*
Vulnerability
-attribute
Added in 2.2
1..*
<<enumeration>>
RelationshipType
relationshipType_contains
relationshipType_containedBy
relationshipType_generates
relationshipType_generatedFrom
relationshipType_ancestorOf
relationshipType_descendantOf
relationshipType_variantOf
relationshipType_distributionArtifact
relationshipType_patchFor
relationshipType_patchApplied
relationshipType_copyOf
relationshipType_fileAdded
relationshipType_fileDeleted
relationshipType_fileModified
relationshipType_dynamicLink
relationshipType_staticLink
relationshipType_dataFile
relationshipType_testcaseOf
relationshipType_buildToolOf
relationshipType_documentation
relationshipType_optionalComponentOf
relationshipType_metafileOf
relationshipType_packageOf
relationshipType_amendment
relationshipType_other
relationshipType_describes
relationshipType_describedBy
relationshipType_prerequisiteFor
relationshipType_hasPrerequisite
CreationInfo
<<Property>> -comment xsd : string
<<Property>> -created dateTimeStamp
<<Property>> -creator xsd:string
<<Property>> -licenseListVersion xsd:string
EcosystemInformation
-committerQuality xs : information
-attribute
Added in 2.2 - to be considered, maybe 3.0? Also consider CHAOSS
Package
<<Property>> -checksum Checksum
<<Property>> -description xsd:string
<<Property>> -downloadLocation xsd:string
<<Property>> -homepage xsd:string
<<Property>> -licenseDeclared AnyLicenseInfo
<<Property>> -originator xsd:string
<<Property>> -packageFileName xsd:string
<<Property>> -sourceinfo xsd:string
<<Property>> -summary xsd:string
<<Property>> -supplier xsd:string
<<Property>> -versionInfo xsd:string
<<Property>> -packageVerificationCode PackageVerificationCode
<<Property>> -filesAnalyzed xsd : boolean
<<Property>> -externalRef : ExternalRef
<<Property>> -semanticVersion SemanticVersion
Snippet
<<Property>> -snippetFromFileFile
<<Property>> -byteRangeStartEnd...
1
1
byteRange
1
StartEndPointer
<<Property>> -startPointe...
<<Property>> -endPointer
externalRef
ExternalRef
<<Property>> -category : ...
<<Property>> -comment : ...
<<Property>> -type : Exte...
<<Property>> -locator : st...
0..1
SemanticVersion
-apiName xsd : string
-majorVersion xsd : string
-minorVersion xsd : string
-patchVesion xsd : string
-prereleaseVersion xsd : string
-buildMetadata BuildMetadata
hasFile
snippetFromFile
Added in 2.2
1
ExternalDocumentRef
<<Property>> -checksum Checksum
<<Property>> -spdxDocument SpdxDocument
<<Property>> -exteranalDocumentId xsd:string
1
1..*
Checksum
<<Property>> -algorithmChecksumAlgorithm
<<Property>> -checksumValue xsd:string
1..*
0..1
1..*
0..*
1
File
<<Property>> -fileType FileType
<<Property>> -checksum Checksum
<<Property>> -fileContributorxsd:string
<<Property>> -fileName xsd:string
<<Property>> -noticeText xsd:string
<<Property>> -artifactOf Project
1
<<enumeration>>
ChecksumAlgorithm
checksumAlgorithm_sha1
checksumAlgorithm_sha256
checksumAlgorithm_md5
<<enumeration>>
ExternalRefType
externalRefType_security
externalRefType_packageManagers
externalRefType_other
OrLaterOperator
<<Property>> -license SimpleLicensingInfo
AnyLicenseInfo
1
PackageVerificationCode
<<Property>> -packageVerificationCodeValue xsd:string
<<Property>> -verificationCodeExcludedValue xsd:string
0..*
2..*
2..*
1
SimpleLicensingInfo
<<Property>> -comment xsd:string
<<Property>> -licenseId xsd:string
<<Property>> -name xsd:string
<<Property>> -seeAlso anyURI
<<Property>> -example xsd:string
ConjunctiveLicenseSet
<<Property>> -member AnyLicenseInfo
DisjunctiveLicenseSet
<<Property>> -member AnyLicenseInfo
0..*
Project
<<Property>> -name xsd:string
<<Property>> -homePage Uri
<<Property>> -uri anyUri
1
WitExceptionOperator
<<Property>> -member SimpleLicensingInfo
<<Property>> -licenseException LicenseException
1
LicenseException
<<Property>> -licenseExceptionId xsd : string
<<Property>> -licenseExceptionText xsd : string
<<Property>> -seeAlso anyUri
<<Property>> -name xsd:string
<<Property>> -example xsd:string
ExtractedLicenseInfo
<<Property>> -extractedText xsd:string
<<enumeration>>
FileType
fileType_source
fileType_binary
fileType_archive
fileType_application
fileType_audio
fileType_image
fileType_text
fileType_video
fileType_documentation
fileType_spdx
fileType_other
License
<<Property>> -isOsiApproved boolean
<<Property>> -licenseText xsd:string
<<Property>> -standardLicenseHeader xsd:string
<<Property>> -standardLicenseTemplate xsd:string
<<Property>> -seeAlso anyUri
<<Property>> -IsDeprecatedLicenseID
<<Property>> -isFsfLibre
2.2 - added isDeprecatedLicenseID and isFsfLibre
ListedLicense

Figure C.1 — SPDX ontology

This figure is licensed under the Creative Commons Attribution 3.0 Unported.

C.2 Agent and tool identifiers

Fields that identify entities that have acted in relation to the SPDX document are single line of text which name the agent or tool and, optionally, provide contact information. For example, "Person: Jane Doe (jane.doe@example.com)", "Organization: ExampleCodeInspect (contact@example.com)" and "Tool: LicenseFind - 1.0". The exact syntax of agent and tool identifications is described below in ABNF.

```
agent                = person / organization

tool                 = "Tool: " name 0*1( " " DASH " " version)

person               = "Person: " name 0*1contact-info

organization         = "Organization: " name 0*1contact-info

name                 = 1*( UNRESERVED ) / U+0022 1*( VCHAR-SANS-QUOTE ) U+002
2

contact-info         = " (" email-addr ")"

email-addr           = local-name-atom *( "." local-name-atom ) "@" domain-na
me-atom 1*( "." domain-name-atom )

version              = 1*VCHAR-SANS-QUOTE

local-name-atom  = 1*( ALPHA / DIGIT /     ; Printable US-ASCII
                       "!" / "#" /         ;  characters not including
                       "$" / "%" /         ;  specials.
                       "&" / "'" /
                       "*" / "+" /
                       "-" / "/" /
                       "=" / "?" /
                       "^" / "_" /
                       "`" / "{" /
                       "|" / "}" /
                       "~" )

domain-name-atom = 1*( ALPHA / DIGIT / "-" )

DASH                 = U+2010 / U+2212 /   ; hyphen, minus, em dash and
                       U+2013 / U+2014     ;  en dash


UNRESERVED           = U+0020-U+0027 /     ; visible unicode characters
                       U+0029-U+0080 /     ;  except '(' and dashes
                       U+00A0-U+200F /
                       U+2011-U+2027 /
                       U+202A-U+2211 /
                       U+2213-U+E01EF


VCHAR-SANS-QUOTE = U+0020-U+0021 /   ; visible unicode characters
                   U+0023-U+0080 /   ;  except quotation mark
                   U+00a0-U+E01EF
```

Annex D (Normative)

SPDX license expressions

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 an SPDX License Expression in this section.

The exact syntax of license expressions is described below in ABNF.

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

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

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

license-ref = ["DocumentRef-"1\*(idstring)":"]"LicenseRef-"1*(idstring)

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

compound-expression = 1*1(simple-expression /


simple-expression "WITH" license-exception-id /

  compound-expression "AND" compound-expression /

  compound-expression "OR" compound-expression ) /

  "(" compound-expression ")" )

license-expression = 1*1(simple-expression / compound-expression)
```

In the following sections we describe in more detail `<license-expression>` construct, a licensing expression string that enables a more accurate representation of the licensing terms of modern-day software.

A valid `<license-expression>` string consists of either:

(i) a simple license expression, such as a single license identifier; or

(ii) a more complex expression constructed by combining smaller valid expressions using Boolean license operators.

There MUST NOT be white space between a license-id and any following `+`. This supports easy parsing and backwards compatibility. There MUST be white space on either side of the operator "WITH". There MUST be white space and/or parentheses on either side of the operators `AND` and `OR`.

In the `tag:value` format, a license expression MUST be on a single line, and MUST NOT include a line break in the middle of the expression.

D.2 Case sensitivity

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

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

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

D.3 Simple license expressions

A simple `<license-expression>` is composed one of the following:

— An SPDX License List Short Form Identifier. For example: CDDL-1.0

— An SPDX License List Short Form Identifier with a unary "+" operator suffix to represent the current version of the license or any later version. For example: CDDL-1.0+

— An SPDX user defined license reference: ["DocumentRef-"1*(idstring)":"]"LicenseRef-"1*(idstring)

Some examples:

```
LicenseRef-23

LicenseRef-MIT-Style-1

DocumentRef-spdx-tool-1.2:LicenseRef-MIT-Style-2
```

D.4 Composite license expressions

D.4.1 Introduction

More expressive composite license expressions can be constructed using "OR", "AND", and "WITH" operators similar to constructing mathematical expressions using arithmetic operators.

For the `tag:value` format, any license expression that consists of more than one license identifier and/or LicenseRef, may optionally be encapsulated by parentheses: "( )".

Nested parentheses can also be used to specify an order of precedence which is discussed in more detail in D.4.5.

D.4.2 Disjunctive "OR" Operator

If presented with a choice between two or more licenses, use the disjunctive binary "OR" operator to construct a new license expression, where both the left and right operands are valid license expression values.

For example, when given a choice between the LGPL-2.1-only or MIT licenses, a valid expression would be:

```
LGPL-2.1-only OR MIT
```

An example representing a choice between three different licenses would be:

```
LGPL-2.1-only OR MIT OR BSD-3-Clause
```

D.4.3 Conjunctive "AND" Operator

If required to simultaneously comply with two or more licenses, use the conjunctive binary "AND" operator to construct a new license expression, where both the left and right operands are valid license expression values.

For example, when one is required to comply with both the LGPL-2.1-only or MIT licenses, a valid expression would be:

```
LGPL-2.1-only AND MIT
```

An example where all three different licenses apply would be:

```
LGPL-2.1-only AND MIT AND BSD-2-Clause
```

D.4.4 Exception "WITH" Operator

Sometimes a set of license terms apply except under special circumstances. In this case, use the binary "WITH" operator to construct a new license expression to represent the special exception situation. A valid `<license-expression>` is where the left operand is a `<simple-expression>` value and the right operand is a `<license-exception-id>` that represents the special exception terms.

For example, when the Bison exception is to be applied to GPL-2.0-or-later, the expression would be:

```
GPL-2.0-or-later WITH Bison-exception-2.2
```

The current set of valid exceptions can be found in A.2. For the most up to date set of exceptions please see spdx.org/licenses. If the applicable exception is not found on the SPDX License Exception List, then use a single `<license-ref>` to represent the entire license terms (including the exception).

D.4.5 Order of precedence and parentheses

The order of application of the operators in an expression matters (similar to mathematical operators). The default operator order of precedence of a `<license-expression>` a is:

```
+
WITH
AND
OR
```

where a lower order operator is applied before a higher order operator.

For example, the following expression:

```
LGPL-2.1-only OR BSD-3-Clause AND MIT
```

represents a license choice between either LGPL-2.1-only and the expression BSD-3-Clause AND MIT because the AND operator takes precedence over (is applied before) the OR operator.

When required to express an order of precedence that is different from the default order a `<license-expression>` can be encapsulated in pairs of parentheses: ( ), to indicate that the operators found inside the parentheses takes precedence over operators outside. This is also similar to the use of parentheses in an algebraic expression e.g., (5+7)/2.

For instance, the following expression:

```
MIT AND (LGPL-2.1-or-later OR BSD-3-Clause)
```

states the OR operator should be applied before the AND operator. That is, one should first select between the LGPL-2.1-or-later or the BSD-3-Clause license before applying the MIT license.

D.4.6 License expressions in RDF

A conjunctive license can be expressed in RDF via a `<spdx:ConjunctiveLicenseSet>` element, with an spdx:member property for each element in the conjunctive license. Two or more members are required.

```
<spdx:ConjunctiveLicenseSet>
    <spdx:member rdf:resource="http://spdx.org/licenses/GPL-2.0-only"/>
    <spdx:ExtractedLicensingInfo rdf:about=
        "http://example.org#LicenseRef-EternalSurrender">
        <spdx:extractedText> In exchange for using this software, you agr
ee to give its author all your worldly possessions. You will not hold the
author liable for all the damage this software will inevitably cause not
only to your person and property, but to the entire fabric of the cosmos.
</spdx:extractedText>
        <spdx:licenseId>LicenseRef-EternalSurrender</spdx:licenseId>
    </spdx:ExtractedLicensingInfo>
</spdx:ConjunctiveLicenseSet>
```

A disjunctive license can be expressed in RDF via a `<spdx:DisjunctiveLicenseSet>` element, with an spdx:member property for each element in the disjunctive license. Two or more members are required.

```
<spdx:DisjunctiveLicenseSet>
    <spdx:member rdf:resource="http://spdx.org/licenses/GPL-2.0-only"/>
    <spdx:member>
        <spdx:ExtractedLicensingInfo rdf:about=
            "http://example.org#LicenseRef-EternalSurrender">
            <spdx:extractedText>In exchange for using this software, you
agree to give its author all your worldly possessions. You will not hold
the author liable for all the damage this software will inevitably cause
not only to your person and property, but to the entire fabric of the cos
mos. </spdx:extractedText>
            <spdx:licenseId>LicenseRef-EternalSurrender</spdx:licenseId>
        </spdx:ExtractedLicensingInfo>
    </spdx:member>
</spdx:DisjunctiveLicenseSet>
```

A License Exception can be expressed in RDF via a `<spdx:LicenseException>` element. This element has the following unique mandatory (unless specified otherwise) attributes:

- `comment` - An `rdfs:comment` element describing the nature of the exception.
- `seeAlso` (optional, one or more)- An `rdfs:seeAlso` element referencing external sources of information on the exception.
- `example` (optional) - Text describing examples of this exception.
- `name` - The full human readable name of the item.
- `licenseExceptionId` - The identifier of an exception in the SPDX License List to which the exception applies.
- `licenseExceptionText` - Full text of the license exception.

```
<rdf:Description rdf:about=
    "http://example.org#SPDXRef-ButIdDontWantToException">
    <rdfs:comment>This exception may be invalid in some jurisdictions.</r
dfs:comment>
    <rdfs:seeAlso>http://dilbert.com/strip/1997-01-15</rdfs:seeAlso>
    <spdx:example>So this one time, I had a license exception…</spdx:exam
ple>
    <spdx:licenseExceptionText>A user of this software may decline to fol
low any subset of the terms of this license upon finding any or all such
terms unfavorable. </spdx:licenseExceptionText>
    <spdx:name>&quot;But I Don&apos;t Want To&quot; Exception</spdx:name>
    <spdx:licenseExceptionId>SPDXRef-ButIdDontWantToException</spdx:licen
seExceptionId>
    <rdf:type rdf:resource="http://spdx.org/rdf/terms#LicenseException"/>
</rdf:Description>
```

Annex E
(Informative)

Using SPDX license list short identifiers in source files

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 read.
- 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.

To the extent that a source file contains existing copyright and license information, it is the SPDX project's recommendation that SPDX short identifiers be used to supplement, not replace that information. When there is a standard header provided by the license author, it is recommended to use such standard header (alone or in combination with the SPDX short identifier). 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. To the extent that the file contains existing license information, it is our recommendation that the tag be used to supplement not replace that information. Of course, this is the ultimate decision of the copyright holders of the file.

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

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

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

E.3 Representing single license

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

Examples:

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

E.4 Representing multiple licenses

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

a) When there is a choice between licenses ("disjunctive license"), they should be separated with "OR". If presented with a choice between two or more licenses, use the disjunctive binary "OR" operator to construct a new license expression.

b) Similarly when multiple licenses need to be simultaneously applied ("conjunctive license"), they should be separated with "AND". If required to simultaneously comply with two or more licenses, use the conjunctive binary "AND" operator to construct a new license expression.

c) In some cases, a set of license terms apply except under special circumstances, in this case, use the "WITH" operator followed by one of the recognized exception identifiers.

d) The expression MUST be on a single line, and MUST NOT include a line break in the middle of the expression.

Examples:

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

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

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

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

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

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

Annex F
(Normative)

External repository identifiers

F.1 Introduction

This specification allows external resources to be referenced from SPDX documents. The identifiers are a combination of a category, a type and a locator.

There are currently four defined categories:

- Security
- Package-Manager
- Persistent-Id
- Other

The following sections provide details on the available types and the locator formats for each of the categories.

F.2 Security

F.2.1 cpe22Type

Locator Format:

```
[c][pP][eE]:/[AHOaho]?(:[A-Za-z0-9\._\-~%]*){0,6}
```

Contextual Example:

```
cpe:/o:canonical:ubuntu_linux:10.04:-:lts
```

External Reference Site: https://nvd.nist.gov/cpe

Documentation: https://cpe.mitre.org/files/cpe-specification_2.2.pdf

F.2.2 cpe23Type

Locator Format:

```
cpe:2\.3:[aho\*\]
(:(((\?*|\*?)([azAZ09\\._]|(\\[\\\*\?!
"#$$%&'\(\)\+,/:;<=>@\[\]\^`\{\|}~])
)+(\?*|\*?))|[\*\])){5}
(:(([azAZ]{2,3}(([azAZ]{2}|[09]{3
}))?)|[\*\]))
```

```
(:(((\?*|\*?)([azAZ09\\._]|(\\[\\\*\?!
"#$$%&'\(\)\+,/:;<=>@\[\]\^`\{\|}~])
)+(\?*|\*?))|[\*\]))){4}
```

Contextual Example:

```
cpe:2.3:o:canonical:ubuntu_linux:10.04::lts:*:*:*:*:*
```

External Reference Site: https://nvd.nist.gov/cpe

Documentation: http://csrc.nist.gov/publications/nistir/ir7695/NISTIR-7695-CPE-Naming.pdf

F.3 Package-Manager

F.3.1 maven-central

Locator Format:

```
group:artifact[:version]
^[^:]+:[^:]+(:[^:]+)?$
```

Contextual Example:

```
org.apache.tomcat:tomcat:9.0.0.M4
```

External Reference Site: http://repo1.maven.org/maven2/

Documentation: https://maven.apache.org

F.3.2 npm

Locator Format:

```
package@version
^[^@]+@[^@]+$
```

Contextual Example:

```
http-server@0.3.0
```

External Reference Site: https://www.npmjs.com/

Documentation: https://docs.npmjs.com/files/package.json

F.3.3 nuget

Locator Format:

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

Contextual Example:

```
Microsoft.AspNet.MVC/5.0.0
```

External Reference Site: https://www.nuget.org/

Documentation: https://docs.nuget.org/

F.3.4 bower

Locator Format:

```
package#version
^[^#]+#[^#]+$
```

Contextual Example:

```
modernizr#2.6.2
```

External Reference Site: http://bower.io/

Documentation: http://bower.io/docs/api/#install

F.3.5 purl

Locator Format:

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

Contextual Example:

```
pkg:docker/debian@sha256:2f04d3d33b6027bb74ecc81397abe780649ec89f1a2af18d
7022737d0482cefe
```

External Reference Site: https://github.com/package-url/purl-spec

Documentation: https://github.com/package-url/purl-spec

F.4 Persistent-Id

F.4.1 swh

These point to objects present in the Software Heritage archive by the means of SoftWare Heritage persistent Identifiers (SWHID), that are guaranteed to remain stable (persistent) over time. Their syntax is described below. Note that they are identifiers and not URLs.

A persistent identifier can point to any software artifact (or "object") available in the Software Heritage archive. Objects come in different types, and most notably:

- contents
- directories

- revisions
- releases
- snapshots

The SWHID follow the `swh:` IANA-registered URI scheme.

Grammar for locator format:

```
<locator> ::= "swh" ":" <scheme_version> ":" <object_type> ":" <object_id
> ;
<scheme_version> ::= "1" ;
<object_type> ::= "cnt" | "dir" | "rev" | "rel" | "snp" ;
<object_id> ::= 40 * <hex_digit> ;  *intrinsic object id, as hex-encoded
SHA1*
<hex_digit> ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
| "a" | "b" | "c" | "d" | "e" | "f" ;
```

Examples:

- `swh:1:cnt:94a9ed024d3859793618152ea559a168bbcbb5e2` points to the content of a file containing the full text of the GPL3 license
- `swh:1:dir:d198bc9d7a6bcf6db04f476d29314f157507d505` points to a directory containing the source code of the Darktable photography application as it was at some point on 4 May 2017
- `swh:1:rev:309cf2674ee7a0749978cf8265ab91a60aea0f7d` points to a commit in the development history of Darktable, dated 16 January 2017, that added undo/redo supports for masks
- `swh:1:rel:22ece559cc7cc2364edc5e5593d63ae8bd229f9f` points to Darktable release 2.3.0, dated 24 December 2016
- `swh:1:snp:c7c108084bc0bf3d81436bf980b46e98bd338453` points to a snapshot of the entire Darktable Git repository taken on 4 May 2017 from GitHub

External documentation: Software Heritage

F.5 Other

F.5.1 [idstring]

Locator Format:

No spaces, but anything else goes

Annex G
(Normative)

SPDX Lite

G.1 Explanation of SPDX Lite

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.

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.

G.2 Format of SPDX Lite

The SPDX Lite profile is a subset of the SPDX specification. SPDX Lite consists of mandatory fields of the Document Creation and Package Information sections and other basic information. Cardinality of each item is not changed.

The mandatory part of the SPDX document creation information section (which consists of SPDX Version, Data License, SPDX Identifier, Document Name, SPDX Document Namespace, Creator and Created) is used for keeping compatibility with SPDX tools.

The main part of the Package Information (those are Package Name, Package Version, Package File Name, Package Download Location, Package Home Page, Concluded License, Declared License, Comments on License and Copyright Text) is used for exchanging license information.

In the Package Information, Package SPDX Identifier and Files Analyzed are used for keeping compatibility with SPDX tools.

Files Analyzed shall be set to "false" when SPDX Lite is used.

Package Comment can be used to describe additional details, such as compiling options, where a license may change with a different compiling option.

The Other License information section (License Identifier, Extracted Text, License Name and License Comment) is used for exchanging license information for licenses that are not on the SPDX License List.

G.3 Table of SPDX Lite fields

Table G.1 — SPDX Lite fields

#	SPDX subclause	Field name
L1.1	6.1	SPDX Version
L1.2	6.2	Data License
L1.3	6.3	SPDX Identifier
L1.4	6.4	Document Name
L1.5	6.5	SPDX Document Namespace
L1.6	6.8	Creator
L1.7	6.9	Created
L2.1	7.1	Package Name
L2.2	7.2	Package SPDX Identifier
L2.3	7.3	Package Version
L2.4	7.4	Package File Name
L2.5	7.7	Package Download Location
L2.6	7.8	Files Analyzed
L2.7	7.11	Package Home Page
L2.8	7.13	Concluded License
L2.9	7.15	Declared License
L2.10	7.16	Comments on License
L2.11	7.17	Copyright Text
L2.12	7.20	Package Comment
L3.1	10.1	License Identifier
L3.2	10.2	Extracted Text
L3.3	10.3	License Name
L3.4	10.5	License Comment

Annex H
(Informative)

SPDX file tags

H.1 Rationale

SPDX short-form license identifiers using the `SPDX-License-Identifier:` tag, as described in Annex E, provide a mechanism for developers to easily convey information about the licenses they declare on a file-by-file basis. That mechanism is intentionally very easy for software tools to identify and detect, since it includes a standard text string that is unlikely to occur in other contexts, and since it uses license identifiers from the SPDX License List or from user-defined `LicenseRef-` statements.

The SPDX specification defines various other fields in the File Information section (Clause 8) that are also useful for conveying information on a file-by-file basis. For example, the REUSE Software guidelines community expressed interest in having a similar method to recommend that developers use to express their copyright notices in a machine-readable manner.

This appendix describes a mechanism, similar to `SPDX-License-Identifier`, for developers to convey such other file-based information easily in comments in their files. This in turn enables software tools to easily find and extract that information, and to insert it into the corresponding fields of an SPDX document generated by those tools.

H.2 Format

An SPDX file tag consists of a single line, generally as part of a comment near the top of the file, in the following format:

```
SPDX-tagname: <value>
```

where *tagname* is replaced by the 'tag' defined for tag-value SPDX documents for that field, according to the File Information section of the SPDX specification. The meaning and semantics of any SPDX file tag are intended to be identical to those described in the File Information section of the SPDX specification.

Examples:

File type (see 8.3):

```
SPDX-FileType: SOURCE
SPDX-FileType: DOCUMENTATION
SPDX-FileType: TEXT
```

Copyright text (see 8.8):

```
SPDX-FileCopyrightText: 2019 Jane Doe <jane@example.com>
SPDX-FileCopyrightText: Copyright 2008-2010 John Smith
SPDX-FileCopyrightText: Copyright Example Company
SPDX-FileCopyrightText: Copyright contributors to the Foo project.
```

File contributors (see 8.14):

```
SPDX-FileContributor: Modified by Jane Doe
SPDX-FileContributor: The Regents of the University of California
```

SPDX file tags of a particular type may appear one or multiple times in a file, depending on the corresponding cardinality defined for that field in the File Information section of the SPDX specification.

Multiple-line values are not recommended, because doing so will make it harder for simple search tools to extract all data by looking only for lines beginning with the relevant tag.

Version 3.0 of the REUSE Software guidelines implements this format, via a recommendation to use the tag `SPDX-FileCopyrightText:` to include copyright notices as part of a file's comment headers.

H.3 Caveats

A creator of an SPDX document may elect to disregard any or all file tags in any file. SPDX document creators should determine for themselves the extent to which they will rely upon the information specified in a file tag.

Not all fields in the File Information section will be useful or relevant to use as file tags. For example, `SPDX-FileName` is unnecessary as it can be easily derived from the actual file name; `SPDX-SPDXID` is likely to be ignored by an SPDX document creator who may need to define their own SPDXRef- ID system for their document; etc.

The short-form license identifiers described in Annex E do not follow the file tag convention described above. The `SPDX-License-Identifier` emerged from the broader community prior to being defined in the SPDX specification, so it does not map to a `License-Identifier` field in the File Information section.

Annex I
(Informative)

Differences from previous editions

I.1 Differences between V2.2.1 and V2.2

There were no technical differences; V2.2.1 is V2.2 reformatted for submission to ISO via the PAS process. As a result, new clauses were added causing the previous clause-numbering sequence to change. Also, Annexes went from having Roman numbers to Latin letters. Here is the translation between numbering in the current and previous editions:

Table I.1 — SPDX Editions

Current edition	Title	V2.2
Clause 1	Scope	N/A
Clause 2	Normative references	N/A
Clause 3	Terms and definitions	N/A
Clause 4	Conformance	N/A
Clause 5	Composition of an SPDX document	N/A
Clause 6	SPDX document creation information section	Chapter 2
Clause 7	Package information section	Chapter 3
Clause 8	File information section	Chapter 4
Clause 9	Snippet information section	Chapter 5
Clause 10	Other licensing information detected section	Chapter 6
Clause 11	Relationships between SPDX elements section	Chapter 7
Clause 12	Annotations section	Chapter 8
Clause 13	Review information section (deprecated)	Chapter 9
Annex A	SPDX license list	Appendix I
Annex B	License matching guidelines and templates	Appendix II
Annex C	RDF object model and identifier syntax	Appendix III
Annex D	SPDX license expressions	Appendix IV
Annex E	Using SPDX short identifiers in source files	Appendix V

Current edition	Title	V2.2
Annex F	External repository identifiers	Appendix VI
Annex G	SPDX Lite	Appendix VIII
Annex H	SPDX file tags	Appendix IX
Annex I	Differences from previous editions	N/A

I.2 Differences from V2.2 and V2.1

— JSON, YAML, and a development version of XML have been added as supported file formats.

— A new appendix "SPDX File Tags" has been added to describe a method that developers can use to document other SPDX file-specific information (such as copyright notices, file type, etc.) in a standardized and easily machine-readable manner. See Appendix IX for more information.

— A new appendix "SPDX Lite" has been added to document a lightweight subset of the SPDX specification for scenarios where a full SPDX document is not required. See Appendix VIII for more information.

— Additional relationship options have been added to enable expression of different forms of dependencies between SPDX elements. As well, NONE and NOASSERTION keywords are now permitted to be used with relationships to indicate what is unknown.

— Miscellaneous bug fixes and non-breaking improvements as reported on the mailing list and reported as issues on the spdx-spec GitHub repository.

I.3 Differences between V2.1 and V2.0

— Snippets have been added to allow a portion of a file to be identified as having different properties from the file it resides in. The use of snippets is completely optional and it is not mandatory for snippets to be identified. See section 5 Snippet Information for further details on the fields available to describe snippets.

— External Packages can now be referred to in SPDX documents. When there is no SPDX file information available to document the content of these external packages, then the filesAnalyzed attribute on a package should be set to false. See section 3.8 Files Analyzed for more information.

— Packages are now able to associate with an “External Reference” which allows a Package to reference an external source of additional information, metadata, enumerations, asset identifiers, or downloadable content believed to be relevant to the Package. See: section 3.21 External Reference, 3.22 External Reference Comment and Appendix VI: External Repository Identifiers for more information.

— The “Artifact of Project” fields at the file level are now deprecated, as they can be replaced by a relationship to the more descriptive External Packages.

— A new appendix “Using SPDX short identifiers in Source Files” has been added to document the best practices to refer to the licenses in the SPDX license list that have emerged from the development community. See Appendix V: Using SPDX short identifiers in Source Files for more information.

- Miscellaneous bug fixes.

I.4 Differences between V2.0 and V1.2

- Abstraction has been applied to the underlying model with the inclusion of SPDX elements. With SPDX 2.0, the concept of an SPDX element is introduced (see Appendix III). This includes SPDX documents, SPDX files, and SPDX packages, each of which gets associated with an SPDX identifier which is denoted by "SPDXRef-".

- SPDX relationships have been added to allow any SPDX element to have a relationship to other SPDX elements. Documented the origin of an SPDX hierarchy of sub-packages, documenting the origin of an SPDX element, and documenting modifications or corrections (annotations) to an SPDX element.

- The ability to reference SPDX elements outside the current SPDX document itself (external references).

- Additional file types are now supported.

- Additional checksum algorithms are now supported.

- Review Information section is deprecated. It is recommended to provide document reviews with Annotations (Section 7).

- A License Expression Syntax has been introduced and documented in Appendix IV.

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, and 2.2; SPDX.dev, https://spdx.dev/specifications

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

ICS 35.080

Price based on 145 pages