Software Package Data Exchange (SPDX®)

Specification Version 2.1.1

Table of Contents

The Software Package Data Exchange (SPDX®) Specification Version 2.1.1

With thanks to Adam Cohn, Andrew Back, Ann Thornton, Bill Schineller, Bruno Cornec, Ciaran Farrell, Daniel German, David Wheeler, Debra McGlade, Dennis Clark, Ed Warnicke, Eran Strod, Eric Thomas, Esteban Rockett, Gary O'Neall, Guillaume Rousseau, Hassib Khanafer, Jack Manbeck, Jaime Garcia, Jeff Luszcz, Jilayne Lovejoy, John Ellis, Karen Copenhaver, Kate Stewart, Kevin Mitchell, Kim Weins, Kirsten Newcomer, Kris Reeves, Liang Cao, Marc-Etienne Vargenau, Mark Gisi, Marshall Clow, Martin Michlmayr, Martin von Willebrand, Matt Germonprez, Michael J. Herzog, Michel Ruffin, Nuno Brito, Oliver Fendt, Paul Madick, Peter Williams, Phil Robb, Philip Odence, Philip Koltun, Phillippe Ombredanne, Pierre Lapointe, Rana Rahal, Robin Gandhi, Sam Ellis, Sameer Ahmed, Scott K Peterson, Scott Lamons, Scott Sterling, Shane Coughlan, Steve Cropper, Stuart Hughes, Tom Callaway, Tom Vidal, Thomas F. Incorvia, Thomas Steenbergen, Venkata Krishna, W. Trevor King, Yev Bronshteyn, and Zachary McFarland for their contributions and assistance.

1 Rationale

1.1 Charter

To create a set of data exchange standards that enable companies and organizations to share license and component information (metadata) for software packages and related content with the aim of facilitating license and other policy compliance.

1.2 Definition

The Software Package Data Exchange (SPDX®) specification is a standard format for communicating the components, licenses, and copyrights associated with software packages. An SPDX file is associated with a particular software package or set of packages and contains information about it in the SPDX format.

1.3 Why is a common format for data exchange needed?

Companies and organizations (collectively "Organizations") are widely using and reusing open source and other software packages. 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. The SPDX working group seeks to create a data exchange format so that information about software packages and related content may be collected and shared in a common format with the goal of saving time and improving data accuracy.

1.4 What does this specification cover?

1.4.1 SPDX Document Creation Information: Meta data to associate analysis results with a specific version of the SPDX file and license for use, and provide information on how, when, and by whom the SPDX file was created.

1.4.2 Package Information: Facts that are common properties of the entire package.

1.4.3 File Information: Facts that are specific to each file included in the package.

1.4.4 Snippet Information: Facts that are specific to only a part of a file.

1.4.5 Other Licensing Information Detected: A way to capture information about and refer to licenses that are not on the SPDX License List.

1.4.6 Relationships Between SPDX Elements: Information on how Documents, Packages & Files relate to each other.

1.4.7 Annotations: Information about when and by whom the SPDX file was reviewed

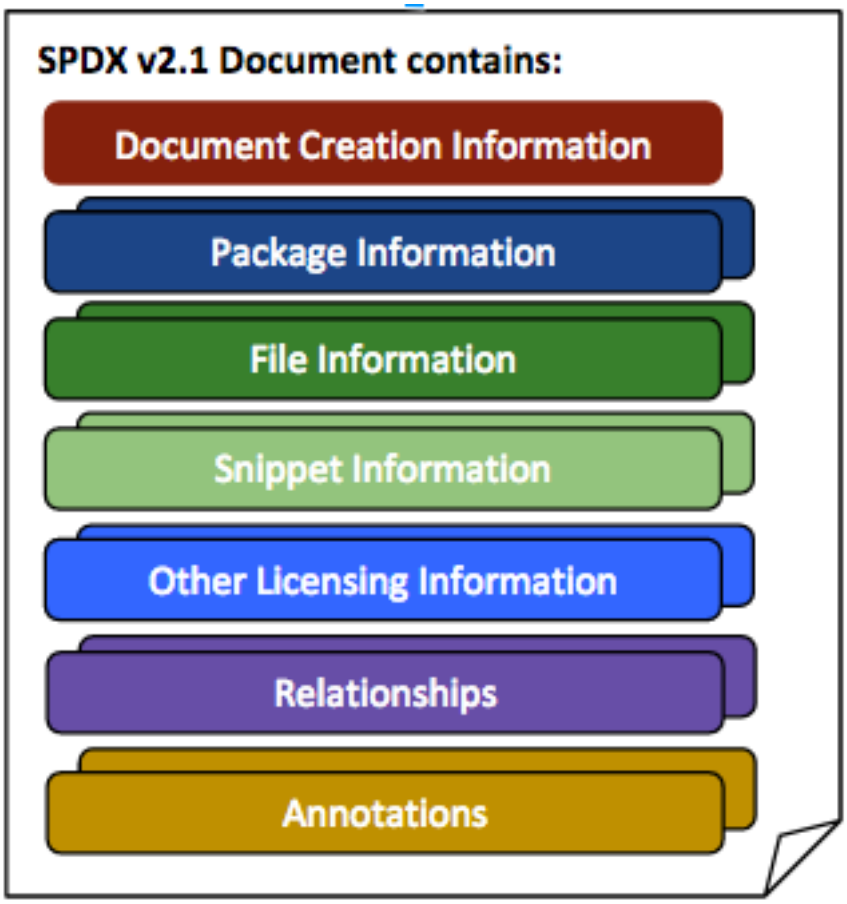

Figure 1: Overview of SPDX 2.1 document contents

1.5 What is not covered in the specification?

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

1.5.2 How the data stored in an SPDX file is used by the recipient.

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

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

1.5.5 Examples may contain ... which indicate detailed text specific to the SPDX Document

1.6 Format Requirements

1.6.1 Must be in a human readable form.

1.6.2 Must be in a syntax that a software tool can read and write.

1.6.3 Must be suitable to be checked for syntactic correctness independent of how it was generated (human or tool).

1.6.4 The SPDX file character set must support UTF-8 encoding.

1.6.5 Must permit automated specification syntax validation.

1.6.6 Resource Description Framework (RDF) can be used to represent this information, as can an annotated tag value flat text file.

1.6.7 Interoperability with an annotate tag:value format and the RDF format will be preserved.

1.6.8 Tags and RDF properties are case sensitive.

1.6.9 Should be easy to recognize in a file system without opening the file. A suggested naming convention is to use *.spdx (for tag:value format) and *-spdx.rdf for RDF format.

1.6.10 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.

1.7 Conformance

1.7.1 A file can be designated an SPDX document, if it is compliant with the requirements of the SPDX Trademark License (See the SPDX Trademark Page).

1.7.2 The official copyright notice to be used with any verbatim reproduction and/or distribution of this SPDX Specification 2.1.1 is:

"Official SPDX® Specification 2.1.1 Copyright © 2010-2018 Linux Foundation and its Contributors. Licensed under the Creative Commons Attribution License 3.0 Unported. All other rights are expressly reserved."

1.7.3 The official copyright notice to be used with any non-verbatim reproduction and/or distribution of this SPDX Specification, 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 2.1.1 found at spdx.org. These portions are Copyright © 2010-2018 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."

1.8 Differences from SPDX Specification 2.0

1.8.1 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 manditory for snippets to be identified. See section 5 snippet information for further details on the fields available to describe snippets.

1.8.2 External Packages can now be refered 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.

1.8.3 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.

1.8.4 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.

1.8.5 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.

1.8.6 Miscellaneous bug fixes as reported on the mailing list and reported as issues on the spdx-spec GitHub repository.

2 Document Creation Information

One instance is required for each SPDX file produced. It provides the necessary information for forward and backward compatibility for processing tools.

Cardinality: Mandatory, one.

Fields:

2.1 SPDX Version

2.1.1 Purpose: 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

will be incremented when incompatible changes between versions are made (one or more sections are created, modified or deleted). The minor field will be incremented when backwards compatible changes are made.

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

2.1.3 Cardinality: Mandatory, one.

2.1.4 Data Format: SPDX—M.N where:

M is major version number

N is minor version number.

2.1.5 Tag: SPDXVersion:

Example:

```
SPDXVersion: SPDX—2.1
```

2.1.6 RDF: spdx:specVersion

Example:

```
<SpdxDocument rdf:about="...">
    <specVersion>SPDX—2.1</specVersion>
</SpdxDocument>
```

2.2 Data License

2.2.1 Purpose: Compliance with the SPDX specification includes populating the SPDX fields therein with data related to such fields ("SPDX-Metadata"). The SPDX specification 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. By using the SPDX specification, or any portion hereof, you hereby agree that any copyright rights (as determined by your jurisdiction) in any SPDX-Metadata, including without limitation explanatory text, shall be subject to the terms of the Creative Commons CC0 1.0 Universal license. For SPDX-Metadata not containing any copyright rights, you hereby agree and acknowledge that the SPDX-Metadata is provided to you "as-is" and without any representations or warranties of any kind concerning the SPDX-Metadata, express, implied, statutory or otherwise, including without limitation warranties of title, merchantability, fitness for a particular purpose, non-infringement, or the absence of latent or other defects, accuracy, or the presence or absence of errors, whether or not discoverable, all to the greatest extent permissible under applicable law.

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

2.2.3 Cardinality: Mandatory, one.

2.2.4 Data Format: CC0—1.0

2.2.5 Tag: DataLicense:

Example:

```
DataLicense: CC0—1.0
```

2.2.6 RDF: spdx:dataLicense

Example:

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

2.3 SPDX Identifier

2.3.1 Purpose: 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 the "Relationships between SPDX Elements" section for examples.

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

2.3.3 Cardinality: Mandatory, one.

2.3.4 DataFormat: SPDXRef—DOCUMENT

2.3.5 Tag: SPDXID:

Example:

```
SPDXID: SPDXRef—DOCUMENT
```

2.3.6 RDF: The URI for the document is the document namespace appended by

#SPDXRef—DOCUMENT

```
<spdx:SpdxDocument rdf:about="http://spdx.org/spdxdocs/spdx-example-444504E0-4F89-41D3-9A0C-0305E82C33...">
 ...
</spdx:SpdxDocument>
```

2.4 Document Name

2.4.1 Purpose: Identify name of this document as designated by creator.

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

2.4.3 Cardinality: Mandatory, one.

2.4.4 DataFormat: Single line of text.

2.4.5 Tag: DocumentName:

Example:

```
DocumentName: glibc-v2.3

DocumentName: ubuntu-14.04
```

2.4.6 RDF: Property spdx:name in class Document

Example:

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

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

2.5 SPDX Document Namespace

2.5.1 Purpose: 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 cannot 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 must 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 must be used. There can only be one URI for an SPDX document and only one SPDX document for a given URI.

2.5.2 Intent: The URI provides an unambiguous mechanism for other SPDX documents to reference SPDX elements within this SPDX document. See section 2.6 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 section 2.4.
- 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 that 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.

2.5.3 Cardinality: Mandatory, one.

2.5.4 Data 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 must contain a scheme (e.g. https:).

The URI must 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 must be used. There can only be one URI for an SPDX document and only one SPDX document for a given URI.

2.5.5 Tag: DocumentNamespace:

Example:

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

2.5.6 RDF: The unique ID is the URI for the SPDX document

Example:

```
<SpdxDocument rdf:about="http://spdx.org/spdxdocs/spdx-tools-v1.2-3F2504E0-4F89-41D3-9A0C-0305
    E82...">
  <rdfs:comment>This document was created using SPDX 2.0 using licenses from the web site.</rdfs:
    comment>
</SpdxDocument>
```

2.6 External Document References

2.6.1 Purpose: Identify any external SPDX documents referenced within this SPDX document.

2.6.2 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.

2.6.3 Cardinality: Optional, one or many.

2.6.4 Data 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 section 2.5 of that referenced document,

[Checksum] is a checksum of the external document following the checksum

format defined in section 3.9.

2.6.5 Tag: ExternalDocumentRef:

Example:

```
ExternalDocumentRef:DocumentRef-spdx-tool-1.2 http://spdx.org/spdxdocs/spdx-tools- v1.2-3F2504E0-4
    F89-41D3-9A0C-0305E82C3301 SHA1: d6a770ba38583ed4bb4525bd96e50461655d2759
```

2.6.6 RDF: Property spdx:externalDocumentRef in class spdx:Document **range** ExternalDocumentRef.

The ExternalDocumentRef contains two properties:

- spdxDocument - the SpdxDocument being referenced

- checksum - the checksum of the referenced SPDX document

Example:

```
<externalDocumentRef>
    <ExternalDocumentRef>
        <spdx:externalDocumentId>DocumentRef—spdx—tool—1.2</spdx:externalDocumentId>
        <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>
```

Notes: in RDF, a namespace can be created for the external document reference if a short form name for the external reference is desired.

2.7 License List Version

2.7.1 Purpose: An optional field for creators of the SPDX file to provide the version of the SPDX License List used when the SPDX file was created.

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

2.7.3 Cardinality: Optional, one.

2.7.4 Data Format: M.N

where:

M is major version number N is minor version number.

2.7.5 Tag: LicenseListVersion :

Example:

```
LicenseListVersion : 2.0
```

2.7.6 RDF: Property licenseListVersion in class spdx:CreationInfo

Example:

```
<CreationInfo>
    < licenseListVersion >2.0</ licenseListVersion >
</CreationInfo>
```

2.8 Creator

2.8.1 Purpose: Identify who (or what, in the case of a tool) created the SPDX file. If the SPDX file was created by an individual, indicate the person's name. If the SPDX file was created on behalf of a company or organization, indicate the entity name. If the SPDX file 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.

2.8.2 Intent: Here, the generation method will assist the recipient of the SPDX file in assessing the general reliability/accuracy of the analysis information.

2.8.3 Cardinality: Mandatory, one or many.

2.8.4 Data Format: Single line of text with the following keywords:

```
"
Person: person "name and optional "(email)"
"Organization: "organization and optional "(email)"
"Tool: toolidentifier —"version
```

2.8.5 Tag: Creator:

Example:

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

2.8.6 RDF: Property spdx:creator in class spdx:CreationInfo

Example:

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

2.9 Created

2.9.1 Purpose: Identify when the SPDX file 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 section 8, which involves the addition of information during a subsequent review.

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

2.9.3 Cardinality: Mandatory, one.

2.9.4 Data 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

2.9.5 Tag: Created:

Example:

```
Created: 2010—01—29T18:30:22Z
```

2.9.6 RDF: Property spdx:created in class spdx:CreationInfo

Example:

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

2.10 Creator Comment

2.10.1 Purpose: An optional field for creators of the SPDX file to provide general comments about the creation of the SPDX file or any other relevant comment not included in the other fields.

2.10.2 Intent: Here, the intent is to provide recipients of the SPDX file with comments by the creator of the SPDX file.

2.10.3 Cardinality: Optional, one.

2.10.4 Data 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>.

2.10.5 Tag: CreatorComment:

Example:

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

2.10.6 RDF: Property rdfs:comment in class spdx:CreationInfo

Example:

```
<CreationInfo>
    <rdfs:comment>This SPDX file 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>
```

2.11 Document Comment

2.11.1 Purpose: An optional field for creators of the SPDX file content to provide comments to the consumers of the SPDX document.

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

2.11.3 Cardinality: Optional, one.

2.11.4 Data 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>.

2.11.5 Tag: DocumentComment:

Example:

```
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.Approved.Licenses.spdx
    .</text>
```

2.11.6 RDF: Property rdfs:comment in class SpdxDocument

Example:

```
<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>
```

3 Package Information

One instance of the Package Information is required per package being described. A package can contain sub-packages, but the information in this section is a reference to the entire contents of the package listed. Starting with SPDX 2.0, it is not necessary to have a package wrapping a set of files.

Cardinality: Optional, one or many.

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 field. All Package Information fields must be grouped together before the start of a Files section, if file(s) are present. All files contained in a package must immediately follow the applicable Package Information. A new Package Information section (via Package Name) denotes the start of another package. Sub-packages should not be nested inside a Package Information section, but should be separate and should use a Relationship to clarify. Annotations and Relationships for the package may appear after the Package Information before any file information.

Fields:

3.1 Package Name

3.1.1 Purpose: Identify the full name of the package as given by the Package Originator.

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

3.1.3 Cardinality: Mandatory, one.

3.1.4 Data Format: Single line of text.

3.1.5 Tag: PackageName:

Example:

```
PackageName: glibc
```

3.1.6 RDF: property spdx:name in class spdx:Package

Example:

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

3.2 Package SPDX Identifier

3.2.1 Purpose: 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.

3.2.2 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.

3.2.3 Cardinality: Mandatory, one.

3.2.4 Data Format: "SPDXRef-"[idstring]

where [idstring] is a unique string containing letters, numbers, ., and/or –.

3.2.5 Tag: SPDXID:

Example:

```
SPDXID: SPDXRef-1
```

3.2.6 RDF: The URI for the element will follow the form:

```
[SPDX DocumentNamespace]#[SPDX Identifier]
```

See section 2.5 for the definition of the SPDX Document Namespace and section 2.3 for the definition of the SPDX Identifier

Example using xml:base:

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

Example using document URI:

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

3.3 Package Version

3.3.1 Purpose: Identify the version of the package.

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

3.3.3 Cardinality: Optional, one.

3.3.4 Data Format: Single line of text.

3.3.5 Tag: PackageVersion:

Example:

```
PackageVersion:  2.11.1
```

3.3.6 RDF: property spdx:versionInfo in class spdx:Package

Example:

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

3.4 Package File Name

3.4.1 Purpose: 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.

3.4.2 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 subdirectory, is being treated as a package, the subdirectory name may be appropriate to provide. Subdirectory name is preceeded with a ./. See RFC 3986 for syntax.

3.4.3 Cardinality: Optional, one.

3.4.4 Data Format: Single line of text.

3.4.5 Tag: PackageFileName:

Example:

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

Example (subdirectory being treated as a package):

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

3.4.6 RDF: property spdx:packageFileName in class spdx:Package

Example:

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

Example (subdirectory being treated as a package):

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

3.5 Package Supplier

3.5.1 Purpose: Identify the actual distribution source for the package/directory identified in the SPDX file. This may or may not be different from the originating distribution source for the package. The name of the Package Supplier must 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:

(i) the SPDX file creator has attempted to but cannot reach a reasonable objective determination;

(ii) the SPDX file creator has made no attempt to determine this field; or

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

3.5.2 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 file or the contents of the package it documents.

3.5.3 Cardinality: Optional, one.

3.5.4 Data Format: Single line of text with the following keywords | NOASSERTION

- Person: person name and optional (<email>)
- Organization: organization name and optional (<email>)

3.5.5 Tag: PackageSupplier:

Example:

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

3.5.6 RDF: property spdx:supplier in class spdx:Package

Example:

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

3.6 Package Originator

3.6.1 Purpose: If the package identified in the SPDX file originated from a different person or organization than identified as Package Supplier (see section 3.5 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 file identifies the package glibc and [Red Hat][RedHat] as the Package Supplier, but the Free Software Foundation is the Package Originator.

Use NOASSERTION if:

(i) the SPDX file creator has attempted to but cannot reach a reasonable objective determination;

(ii) the SPDX file creator has made no attempt to determine this field; or

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

3.6.2 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 file or the contents of the Package it documents.

3.6.3 Cardinality: Optional, one.

3.6.4 Data Format: Single line of text with the following keywords | NOASSERTION

- Person: person name and optional (<email>)
- Organization: organization name and optional (<email>)

3.6.5 Tag: PackageOriginator:

Example:

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

3.6.6 RDF: property spdx: originator in class spdx:Package

Example:

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

3.7 Package Download Location

3.7.1 Purpose: 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 file was created.

Use:

- NONE if there is no download location whatsoever.
- NOASSERTION if:

  (i) the SPDX file creator has attempted to but cannot reach a reasonable objective determination;

  (ii) the SPDX file creator has made no attempt to determine this field; or

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

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

3.7.3 Cardinality: Mandatory, one.

3.7.4 Data 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 must 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.

3.7.5 Tag: PackageDownloadLocation:

Examples if ambiguous:

```
PackageDownloadLocation: NOASSERTION

PackageDownloadLocation: NONE
```

Example for a plain URL:

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

Example 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
```

```
6
7  PackageDownloadLocation: git+ssh:// git .myproject.org/MyProject. git
8
9  PackageDownloadLocation: git+git :// git .myproject.org/MyProject
10
11 PackageDownloadLocation: git+git@git.myproject.org:MyProject
```

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

```
1  PackageDownloadLocation: git:// git .myproject.org/MyProject#src/somefile.c
2
3  PackageDownloadLocation: git+https:// git .myproject.org/MyProject#src/Class.java
```

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

```
1  PackageDownloadLocation: git:// git .myproject.org/MyProject.git@master
2
3  PackageDownloadLocation: git+https:// git .myproject.org/MyProject.git@v1.0
4
5  PackageDownloadLocation: git:// git .myproject.org/MyProject.
       git@da39a3ee5e6b4b0d3255bfef95601890afd80709
```

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

```
1  PackageDownloadLocation: git+https:// git .myproject.org/MyProject.git@master#/src/MyClass.cpp
2
3  PackageDownloadLocation: git+https:// git .myproject.org/
       MyProject@da39a3ee5e6b4b0d3255bfef95601890afd80709#lib/variable.rb
```

Example for Mercurial:

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

The supported forms are:

```
1  PackageDownloadLocation: hg+http://hg.myproject.org/MyProject
2
3  PackageDownloadLocation: hg+https://hg.myproject.org/MyProject
4
5  PackageDownloadLocation: hg+ssh://hg.myproject.org/MyProject
```

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

```
1  PackageDownloadLocation: hg+https://hg.myproject.org/MyProject#src/somefile.c
2
3  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
```

Example 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#src/somefile.c

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

You can specify a revision using the @ delimiter:

```
PackageDownloadLocation: svn+https://svn.myproject.org/svn/MyProject/trunk@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@1234#lib/variable/variable.rb
```

Example 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#src/somefile.c

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

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

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

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@2019#src/somefile.c
```

3.7.6 RDF: property spdx:downloadLocation in class spdx:Package

Example:

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

<Package rdf:about="...">
    <downloadLocation>
```

```
7             git+https :// git . myproject.org/MyProject.git@v10.0#src/ lib .c
8         </downloadLocation>
9     </Package>
10
11    <Package rdf:about=" ... ">
12        <downloadLocation rdf:resource="http :// spdx.org/ rdf/terms#noassertion"/>
13    </Package>
14
15    <Package rdf:about=" ... ">
16        <downloadLocation rdf:resource="http :// spdx.org/ rdf/terms#none"/>
17    </Package>
```

3.8 Files Analyzed

3.8.1 Purpose: 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 must not contain any files.

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

Some examples:

A 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. 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 Section 7. File derived from external product: Package A contains multiple files derived from an outside project. Rather than use the artifactOf * attributes (Sections 4.9-4.11) to describe the relation of these files to their project, the outside project can be represented by another package, Package B, whose FilesAnalyzed attribute is set to false. Each of the binary files can then have a relationship to package B (Section 6). 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.

3.8.3 Cardinality: Optional, one. If omitted, the default value of true is assumed.

3.8.4 Data Format: Boolean

3.8.5 Tag: FilesAnalyzed

Example:

```
FilesAnalyzed: false
```

3.8.6 RDF: property spdx: filesAnalyzed in class spdx:Package

Example:

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

3.9 Package Verification Code

3.9.1 Purpose: This field provides an independently reproducible mechanism identifying specific contents of a package based on the actual files (except the SPDX file itself, if it is included in the package) that make up each package and that correlates to the data in this SPDX file. 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 file as part of a package.

3.9.2 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 file refers to. This field also permits embedding the SPDX file within the package without altering the identifier.

3.9.3 Cardinality: Mandatory, one if FilesAnalyzed is true or omitted, zero (must be omitted) if FilesAnalyzed is false.

3.9.4 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 values 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)

3.9.5 Data Format: single line of text with 160 bit binary represented as 40 lowercase hexadecimal digits

3.9.6 Tag: PackageVerificationCode: (and optionally (excludes: FileName))

FileName is specified in section 4.1.

Example:

```
PackageVerificationCode: d6a770ba38583ed4bb4525bd96e50461655d2758 (excludes: ./package.spdx)
```

3.9.7 RDF: spdx:packageVerificationCodeValue, spdx:packageVerificationCodeExcludedFile in class spdx:PackageVerificationCode in class spdx:Package

Example:

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

3.10 Package Checksum

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

3.10.2 Intent: Eliminate confusion over which version or modification of a specific package the SPDX file references by providing a unique identifier of the package.

3.10.3 Cardinality: Optional, one or many.

3.10.4 Algorithms that can be used: SHA1, SHA256, MD5

3.10.5 Data 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).

3.10.6 Tag: PackageChecksum:

Example:

```
PackageChecksum: SHA1: 85ed0817af83a24ad8da68c2b5094de69833983c

PackageChecksum: SHA256: 11b6d3ee554eedf79299905a98f9b9a04e498210b59f15094c916c91d150efcd

PackageChecksum: MD5: 624c1abb3664f4b35547e7c73864ad24
```

3.10.7 RDF: properties spdx:algorithm, spdx:checksumValue in class spdx:checksum in class spdx:Package

Example:

```
<Package rdf:about="...">
    <checksum>
        <Checksum>
            <algorithm rdf:resource="http://spdx.org/rdf/terms#checksumAlgorithm_sha1"/>
            <checksumValue>85ed0817af83a24ad8da68c2b5094de69833983c</checksumValue>
        </Checksum>
    </checksum>
    <checksum>
        <Checksum>
            <algorithm rdf:resource="http://spdx.org/rdf/terms#checksumAlgorithm_sha256"/>
            <checksumValue>
                11b6d3ee554eedf79299905a98f9b9a04e498210b59f15094c916c91d150efcd
            </checksumValue>
        </Checksum>
    </checksum>
    <checksum>
        <Checksum>
            <algorithm rdf:resource="http://spdx.org/rdf/terms#checksumAlgorithm_md5"/>
            <checksumValue>624c1abb3664f4b35547e7c73864ad24</checksumValue>
        </Checksum>
    </checksum>
</Package>
```

3.11 Package Home Page

3.11.1 Purpose: Provide a place for the SPDX file 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 file creator.

Use:

- NONE if there is no package home page whatsoever.
- NOASSERTION if:

  (i) the SPDX file creator has attempted to but cannot reach a reasonable objective determination;

  (ii) the SPDX file creator has made no attempt to determine this field; or

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

3.11.2 Intent: Save the recipient of the SPDX file who is looking for more info from having to search for and verify a match between the package and the associated project homepage.

3.11.3 Cardinality: Optional, one.

3.11.4 Data Format: uniform resource locator | NONE | NOASSERTION

3.11.5 Tag: PackageHomePage:

Example:

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

3.11.6 RDF: property doap:homepage in class spdx:Package

Example:

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

3.12 Source Information

3.12.1 Purpose: Provide a place for the SPDX file 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.

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

3.12.3 Cardinality: Optional, one.

3.12.4 Data Format: free form text that can span multiple lines.

In tag:value format this is delimited by <text >...</ text>.

3.12.5 Tag: PackageSourceInfo:

Example:

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

3.12.6 RDF: spdx:sourceInfo

Example:

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

3.13 Concluded License

3.13.1 Purpose: Contain the license the SPDX file 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 Appendix IV;
- NONE, if the SPDX file creator concludes there is no license available for this package; or
- NOASSERTION if:
  - (i) the SPDX file creator has attempted to but cannot reach a reasonable objective determination;
  - (ii) the SPDX file creator has made no attempt to determine this field; or
  - (iii) the SPDX file 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, a written explanation should be provided in the Comments on License field (section 3.16). With respect to NOASSERTION, a written explanation in the Comments on License field (section 3.16) is preferred.

3.13.2 Intent: Here, the intent is for the SPDX file 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.

3.13.3 Cardinality: Mandatory, one.

3.13.4 Data Format: <SPDX License Expression> | NONE | NOASSERTION

where:

<SPDX License Expression> is a valid SPDX License Expression as defined in Appendix IV.

3.13.5 Tag: PackageLicenseConcluded:

Example:

```
PackageLicenseConcluded: LGPL-2.0
```

Example:

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

3.13.6 RDF: property spdx:licenseConcluded in **class** spdx:Package

Example:

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

Example:

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

3.14 All Licenses Information from Files

3.14.1 Purpose: 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:

  (i) the SPDX file creator has made no attempt to determine this field; or

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

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

3.14.3 Cardinality: Mandatory, one or many if FilesAnalyzed is true or omitted, zero (must be omitted) if FilesAnalyzed is false.

3.14.4 Data Format: < shortIdentifier > | ["DocumentRef-"[idstring]:]"LicenseRef-"[idstring] | NONE | NOASSERTION

where:

- "DocumentRef-"[idstring] is an optional reference to an external SPDX document as described in section 2.6.
- [idstring] is a unique string containing letters, numbers, ., or —.

3.14.5 Tag: PackageLicenseInfoFromFiles:

Example:

```
PackageLicenseInfoFromFiles: GPL—2.0

PackageLicenseInfoFromFiles: LicenseRef—1

PackageLicenseInfoFromFiles: LicenseRef—2
```

3.14.6 RDF: property spdx: licenseInfoFromFiles in class spdx:Package

Example:

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

```
    <licenseInfoFromFiles rdf:resource="#LicenseRef-2" />
    ...
</Package>
```

3.15 Declared License

3.15.1 Purpose: 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 Appendix IV;
- NONE, if the package contains no license information whatsoever; or
- NOASSERTION if:
  (i) the SPDX file creator has made no attempt to determine this field; or
  (ii) the SPDX file creator has intentionally provided no information (no meaning should be implied by doing so).

3.15.2 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 (section 3.13). This field may have multiple Declared Licenses, if multiple licenses are declared at the package level.

3.15.3 Cardinality: Mandatory, one.

3.15.4 Data Format: <SPDX License Expression> | NONE | NOASSERTION

where:

- <SPDX License Expression> is a valid SPDX License Expression as defined in Appendix IV.

3.15.5 Tag: PackageLicenseDeclared:

Example:

```
PackageLicenseDeclared: LGPL-2.0
```

Example:

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

3.15.6 RDF: property spdx:licenseDeclared in class spdx:Package

Example:

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

Example:

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

3.16 Comments on License

3.16.1 Purpose: This field provides a place for the SPDX file 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 file creator. Its is also preferable to include an explanation here when the Concluded License is NOASSERTION.

3.16.2 Intent: Here, the intent is to provide the recipient of the SPDX file 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.

3.16.3 Cardinality: Optional, one.

3.16.4 Data 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 <licenseComment>.

3.16.5 Tag: PackageLicenseComments:

Example:

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

3.16.6 RDF: property spdx:licenseComments in class spdx:Package

Example:

```
<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>
```

3.17 Copyright Text

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

3.17.2 Intent: Record any copyright notices for the package.

3.17.3 Cardinality: Mandatory, one.

3.17.4 Data Format: free form text that can span multiple lines | NONE | NOASSERTION

3.17.5 Tag: PackageCopyrightText:

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

Example:

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

3.17.6 RDF: property spdx:copyrightText in class spdx:Package

Example:

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

3.18 Package Summary Description

3.18.1 Purpose: This field is a short description of the package.

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

3.18.3 Cardinality: Optional, one.

3.18.4 Data Format: free form text that can span multiple lines.

3.18.5 Tag: PackageSummary:

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

Example:

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

3.18.6 RDF: property spdx:summary in class spdx:Package

Example:

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

3.19 Package Detailed Description

3.19.1 Purpose: This field is a more detailed description of the package. It may also be extracted from the packages itself.

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

3.19.3 Cardinality: Optional, one.

3.19.4 Data Format: free form text than can span multiple lines.

3.19.5 Tag: PackageDescription:

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

Example:

```
PackageDescription: <text>The GNU C Library defines functions that are specified 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>
```

3.19.6 RDF: property spdx:description in class spdx:Package

Example:

```
<Package rdf:about=" ... ">
      ...
      <description>
          The GNU C Library defines functions that are specified 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.
      </description>
      ...
</Package>
```

3.20 Package Comment

3.20.1 Purpose: This field provides a place for the SPDX file creator to record any general comments about the package being described.

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

3.20.3 Cardinality: Optional, one.

3.20.4 Data Format: free form text that can span multiple lines.

3.20.5 Tag: PackageComment:

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

Example:

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

3.20.6 RDF: property rdfs:comment in class spdx:Package

Example:

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

3.21 External Reference

3.21.1 Purpose: 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.

3.21.2 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.

3.21.3 Cardinality: Optional (one or many)

3.21.4 Data Format: <category> <**type**> <locator>

where:

- <category> is SECURITY | PACKAGE—MANAGER | OTHER
- <**type**> is one of the types listed in Appendix VI.

<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**>.

3.21.5 Tag: ExternalRef:

Example:

```
ExternalRef: SECURITY cpe23Type cpe:2.3:a:pivotal_software:spring_framework:4.1.0:*:*:*:*:*:*:*

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

3.21.6 RDF: property target in class spdx:ExternalRef in class spdx:Package

Example (for a listed location):

```
<spdx:Package rdf:about="...">
    ...
    <spdx:externalRef>
        <spdx:ExternalRef>
            <spdx:referenceCategory rdf:resouce="http://spdx.org/rdf/terms#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>
```

Example (for an unlisted location):

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

The referenceType value for a non-listed location consists of the SPDX document namespace (per section 2.5) followed by a # and the category as defined in 3.21.4.

3.22 External Reference Comment

3.22.1 Purpose: To provide human-readable information about the purpose and target of the reference.

3.22.2 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 may need to be explained here. If the reference is BINARY, its relationship to PackageDownloadLocation may need to be explained. If the reference is SOURCE, its relationship to PackageDownloadLocation and SourceInformation may need to be explained.

3.22.3 Cardinality: Conditional (Optional, one) for each [External Reference][#3.21).

3.22.4 Data 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 so that the association can be made.

In RDF, it is delimited by <ExternalRefComment>.

3.22.5 Tag: ExternalRefComment:

Example:

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

3.22.6 RDF: Property rdfs:comment in class spdx:ExternalRef in class spdx:Package

```
<spdx:Package rdf:about=" ... ">
    ...
    <spdx:externalRef>
        <spdx:ExternalRef>
            <spdx:referenceCategory rdf:resouce="http:// spdx.org/ rdf/terms#
    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>
```

4 File Information

One instance of the File Information is required 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.

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 section must follow its Package Information section. If a File is not part of any package, it must precede any Package Information section reference in the SPDX document. The first field to start off the description of a File must 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.

4.1 File Name

4.1.1 Purpose: Identify the full path and filename that corresponds to the file information in this section.

4.1.2 Intent: To aid finding the correct file which corresponds to the file information.

4.1.3 Cardinality: Mandatory, one.

4.1.4 Data 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.

4.1.5 Tag: FileName:

Example:

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

4.1.6 RDF: Property spdx:fileName in class spdx: File

Example:

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

4.2 File SPDX Identifier

4.2.1 Purpose: 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.

4.2.2 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.

4.2.3 Cardinality: Mandatory, one.

4.2.4 DataFormat: "SPDXRef-"[idstring]

where [idstring] is a unique string containing letters, numbers, . and/or –.

4.2.5 Tag: SPDXID:

Example:

```
SPDXID: SPDXRef–1
```

4.2.6 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.

Example using xml:base:

```
<rdf:RDF xml:base="http://acme.com/spdxdocs/acmeproj/v1.2/1BE2A4FF–5F1A–48D3–8483–28A9B0349A1B"
    ...
    <File rdf:ID"=SPDXRef"–1>
        ...
    </File>
```

Example using document URI:

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

4.3 File Type

4.3.1 Purpose: 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 an 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.)

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

4.3.3 Cardinality: Optional, multiple.

4.3.4 Data Format: SOURCE | BINARY | ARCHIVE | APPLICATION | AUDIO | IMAGE | TEXT | VIDEO | DOCUMENTATION | SPDX | OTHER

4.3.5 Tag: FileType:

Example:

```
FileType: BINARY
```

Example: (for a README.TXT)

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

Example (foo.exe)

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

4.3.6 RDF: Property spdx:fileType in class spdx:File

Example:

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

Example: (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>
```

4.4 File Checksum

4.4.1 Purpose: Provide a unique identifier to match analysis information on each specific file in a package.

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

4.4.3 Cardinality: Mandatory, one SHA1, others may be optionally provided.

4.4.4 Algorithm: SHA1() is to be used on the file. Other algorithms that can be provided optionally include SHA256(), MD5().

4.4.5 Data Format: In tag:value there are three components, an algorithm identifier (SHA1), a separator (":") and a checksum value. The RDF must 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.

4.4.6 Tag: FileChecksum:

Example:

```
FileChecksum: SHA1: d6a770ba38583ed4bb4525bd96e50461655d2758

FileChecksum: MD5: 624c1abb3664f4b35547e7c73864ad24
```

4.4.7 RDF: Property spdx:Checksum in class spdx: File

Example:

```
<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>
```

4.5 Concluded License

4.5.1 Purpose: This field contains the license the SPDX file 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 Appendix IV;

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

NOASSERTION, if:

(i) the SPDX file creator has attempted to, but cannot reach a reasonable objective determination;

(ii) the SPDX file creator has made no attempt to determine this field; or

(iii) the SPDX file 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 (section 4.7). With respect to NOASSERTION, a written explanation in the Comments on License field (section 4.7) is preferred.

4.5.2 Intent: Here, the intent is for the SPDX file creator to analyze the License Information in file (section 4.6) 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.

4.5.3 Cardinality: Mandatory, one.

4.5.4 Data Format: <SPDX License Expression> | NONE | NOASSERTION

where:

<SPDX License Expression> is a valid SPDX License Expression as defined in Appendix IV.

4.5.5 Tag: LicenseConcluded:

Example:

```
1 LicenseConcluded: LGPL-2.0
```

Example:

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

4.5.6 RDF: Property spdx:licenseConcluded in class spdx: File

Example:

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

Example:

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

4.6 License Information in File

4.6.1 Purpose: 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 may 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:

(i) the SPDX file creator has made no attempt to determine this field; or

(ii) the SPDX file 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.

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

4.6.3 Cardinality: Mandatory, one or many.

4.6.4 Data Format: <SPDX License Expression> |

[“DocumentRef-”[idstring]“:”]“LicenseRef-”[idstring] |

| NONE | NOASSERTION

where:

<SPDX License Expression> is a valid SPDX License Expression

as defined in Appendix IV.

“DocumentRef-”[idstring]: is an optional reference to an external SPDX

document as described in section 2.6

[idstring] is a unique string containing letters, numbers, . and/or –

4.6.5 Tag: LicenseInfoInFile :

Example:

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

4.6.6 RDF: Property spdx: licenseInfoInFile in class spdx: File

Example:

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

4.7 Comments on License

4.7.1 Purpose: This field provides a place for the SPDX file 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 file creator. It is also preferable to include an explanation here when the Concluded License is NOASSERTION.

4.7.2 Intent: Here, the intent is to provide the recipient of the SPDX file 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.

4.7.3 Cardinality: Optional, one.

4.7.4 Data Format: Free form text that can span multiple lines

4.7.5 Tag: LicenseComments:

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

Example:

```
LicenseComments: <text>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 .</text>
```

4.7.6 RDF: Property spdx:licenseComments in class spdx: File

Example:

```
<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>
```

4.8 Copyright Text

4.8.1 Purpose: Identify the copyright holder of the file, as well as any dates present. This will be a freeform 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

(i) the SPDX document creator has made no attempt to determine this field; or

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

4.8.2 Intent: Record any copyright notice for the file.

4.8.3 Cardinality: Mandatory, one.

4.8.4 Data Format: Free form text that can span multiple lines | NONE | NOASSERTION

4.8.5 Tag: FileCopyrightText :

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

Example:

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

4.8.6 RDF: Property spdx:copyrightText in class spdx: File

Example:

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

##4.9 Artifact of Project Name (deprecated)

4.9.1 Purpose: To indicate that a file has been derived from a specific project.

4.9.2 Intent: To make it easier for recipients of the SPDX file 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.

4.9.3 Cardinality: Optional, one or many.

4.9.4 Data Format: Single line of text. In tag:value format the ArtifactOfProjectName must precede any optional ArtifactOf optional properties (e.g. ArtifactOfHomePage and ArtifactOfURI).

4.9.5 Tag: ArtifactOfProjectName:

Example:

```
ArtifactOfProjectName: Jena
```

4.9.6 RDF: Property spdx: artifactOf /doap:Project/doap:name

Example:

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

4.10 Artifact of Project Homepage (deprecated)

4.10.1 Purpose: To indicate the location of the project from which the file has been derived.

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

4.10.3 Cardinality: Optional, one or many.

4.10.4 Data Format: Uniform Resource Locator | UNKNOWN.

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

4.10.5 Tag: ArtifactOfProjectHomePage:

Example:

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

4.10.6 RDF: spdx: artifactOf /doap:Project/doap:homepage

Example:

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

4.11 Artifact of Project Uniform Resource Identifier (deprecated)

4.11.1 Purpose: To provide a linkage to the project resource in the DOAP document and permit interoperability between the different formats supported.

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

4.11.3 Cardinality: Optional, one or many.

4.11.4 Data Format: Uniform Resource Identifier.

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

4.11.5 Tag: ArtifactOfProjectURI :

Example:

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

4.11.6 RDF: spdx: artifactOf /doap

Example:

```
<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 -->
```

4.12 File Comment

4.12.1 Purpose: This field provides a place for the SPDX file creator to record any general comments about the file.

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

4.12.3 Cardinality: Optional, one.

4.12.4 Data Format: Free form text that can span multiple lines

4.12.5 Tag: FileComment:

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

Example:

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

4.12.6 RDF: Property rdfs:comments in class spdx: File

Example:

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

4.13 File Notice

4.13.1 Purpose: This field provides a place for the SPDX file creator to record license notices or other such related notices found in the file. This may or may not include copyright statements.

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

4.13.3 Cardinality: Optional, one.

4.13.4 Data Format: Free form text that can span multiple lines

4.13.5 Tag: FileNotice :

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

Example:

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

4.13.6 RDF: Property noticeText in class spdx: File

Example:

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

4.14 File Contributor

4.14.1 Purpose: This field provides a place for the SPDX file creator to record file contributors. Contributors could include names of copyright holders and/or authors who may not be copyright holders, yet contributed to the file content.

4.14.2 Intent: Here, the intent is to provide the recipient of the SPDX file 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.

4.14.3 Cardinality: Optional, one or many.

4.14.4 Data Format: Free form text on a single line.

4.14.5 Tag: FileContributor :

In tag:value format single line per contributor.

Example:

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

4.14.6 RDF: Property fileContributor in class spdx:File

Example:

```
<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>
```

4.15 File Dependencies (deprecated)

This field is deprecated since SPDX 2.0 in favor of using Section 7 which provides more granularity about relationships.

4.15.1 Purpose: The field provides a place for the SPDX file creator to record a list of other files (referenceable within this SPDX file) 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 may not necessarily represent the list of all file dependencies, but possibly the ones that impact the licensing and/or may be needed as part of the file distribution obligation.

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

4.15.3 Cardinality: Optional, one or many.

4.15.4 Data Format: Reference to the file within the SPDX document. For the tag:value format, this will be the filename. For the RDF format, it will be a reference to the actual file node.

4.15.5 Tag: FileDependency:

Example:

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

4.15.6 RDF: Property spdx:fileDependency in class spdx: File

Example:

```
<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>
```

5 Snippet Information

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 needs to be associated with a specific File in an SPDX Document.

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

If a File contains Snippets, the Snippet Information section should 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 must 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.

5.1 Snippet SPDX Identifier

5.1.1 Purpose: 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.

5.1.2 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.

5.1.3 Cardinality: Mandatory, one.

5.1.4 DataFormat: SPDXRef—[idstring]

where [idstring] is a unique string containing letters, numbers, . and/or —.

5.1.5 Tag: SnippetSPDXID:

Example:

```
SnippetSPDXID: SPDXRef—1
```

5.1.6 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.

Example using xml:base:

```
<rdf:RDF xml:base="http://acme.com/spdxdocs/acmeproj/v1.2/1BE2A4FF—5F1A—48D3—8483—28A9B0349A1B"
    ...
    <Snippet rdf:ID"=SPDXRef"—1>
        ...
    </Snippet>
```

Example using document URI:

```
<Snippet rdf:about="http://acme.com/spdxdocs/acmeproj/v1.2/1BE2A4FF—5F1A—48D3—8483—28A9B0349...">
    ...
</Snippet>
```

5.2 Snippet from File SPDX Identifier

5.2.1 Purpose: Uniquely identify the file in an SPDX document which this snippet is associated with.

5.2.2 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.

5.2.3 Cardinality: Mandatory, one.

5.2.4 DataFormat: ["DocumentRef-"[idstring]":"] SPDXID

where DocumentRef—[idstring]: is an optional reference to an external

SPDX document as described in section 2.6

where SPDXID is a string containing letters, numbers, . and/or —. as

described in sections (2.3, 3.2, 4.2).

5.2.5 Tag: SnippetFromFileSPDXID:

Example (snippet from a File in local SPDX Doc):

```
SnippetFromFileSPDXID: SPDXRef—filecontainingsnippet
```

Example (snippet from a File in an External SPDX Doc):

```
SnippetFromFileSPDXID: DocumentRef—ExternalDoc1:SPDXRef—filecontainingsnippet
```

5.2.6 RDF: Property spdx:snippetFromFile in class spdx:Snippet

Example (snippet from a File in local SPDX Doc):

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

Example (snippet from a File in an External SPDX Doc):

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

5.3 Snippet Byte Range

5.3.1 Purpose: This field defines the byte range in the original host file (in 5.2) that the snippet information applies to.

5.3.2 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/).

5.3.3 Cardinality: Mandatory, one.

5.3.4 Data 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.

5.3.5 Tag: SnippetByteRange:

Example:

```
SnippetByteRange: 310:420
```

5.3.6 RDF: Property spdx:byteRange 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

Example:

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

<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>
```

5.4 Snippet Line Range

5.4.1 Purpose: This optional field defines the line range in the original host file (in 5.2) 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.

5.4.2 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/).

5.4.3 Cardinality: Optional, one.

5.4.4 Data 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.

5.4.5 Tag: SnippetLineRange:

Example:

```
SnippetLineRange: 5:23
```

5.4.6 RDF: properties spdx:byteRange 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

Example:

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

```
<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>
```

5.5 Snippet Concluded License

5.5.1 Purpose: This field contains the license the SPDX file 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 Appendix IV.

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:

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

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

(iii) the SPDX document creator has made no attempt to determine a Concluded License;

(iv) 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 (section 5.7). With respect to NOASSERTION, a written explanation in the Comments on License field (section 5.7) is preferred.

5.5.2 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 file 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.

5.5.3 Cardinality: Mandatory, one.

5.5.4 Data Format: <SPDX License Expression> | NONE | NOASSERTION

where:

<SPDX License Expression> is a valid SPDX License Expression as defined in Appendix IV.

5.5.5 Tag: SnippetLicenseConcluded:

Example:

```
SnippetLicenseConcluded: GPL-2.0
```

Example:

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

5.5.6 RDF: Property spdx:licenseConcluded in class spdx:Snippet

Example:

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

Example:

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

5.6 License Information in Snippet

5.6.1 Purpose: 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:

(i) the SPDX snippet creator has made no attempt to determine this field; or

(ii) the SPDX snipppet 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.

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

5.6.3 Cardinality: Optional, one or many.

5.6.4 Data Format: <SPDX License Expression> |

["DocumentRef-"[idstring]:"]"LicenseRef-"[idstring] |

NONE | NOASSERTION

where:

<SPDX License Expression> is a valid SPDX License Expression

as defined in Appendix IV.

"DocumentRef-"[idstring]: is an optional reference to an external SPDX

document as described in section 2.6

[idstring] is a unique string containing letters, numbers, . and/or –.

5.6.5 Tag: LicenseInfoInSnippet:

Example:

```
1  LicenseInfoInSnippet: LGPL-2.0
2
3  LicenseInfoInSnippet: LicenseRef-2
```

5.6.6 RDF: Property spdx: licenseInfoInSnippet in class spdx:Snippet

Example:

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

5.7 Snippet Comments on License

5.7.1 Purpose: 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.

5.7.2 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 File, is marked NOASSERTION, or other helpful information relevant to determining the license of the snippet in a file.

5.7.3 Cardinality: Optional, one.

5.7.4 Data Format: Free form text that can span multiple lines

5.7.5 Tag: SnippetLicenseComments:

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

Example:

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

5.7.6 RDF: Property spdx:licenseComments in class spdx:Snippet

Example:

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

5.8 Snippet Copyright Text

5.8.1 Purpose: Identify the copyright holder of the snippet, as well as any dates present. This will 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 file contains no copyright information whatsoever; or

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

5.8.2 Intent: Record any copyright notice associated with the snippet.

5.8.3 Cardinality: Mandatory, one.

5.8.4 Data Format: Free form text that can span multiple lines | NONE | NOASSERTION

5.8.5 Tag: SnippetCopyrightText:

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

Example:

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

5.8.6 RDF: Property spdx:copyrightText in class spdx:Snippet

Example:

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

5.9 Snippet Comment

5.9.1 Purpose: This field provides a place for the SPDX document creator to record any general comments about the snippet.

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

5.9.3 Cardinality: Optional, one.

5.9.4 Data Format: Free form text that can span multiple lines

5.9.5 Tag: SnippetComment:

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

Example:

```
SnippetComment: <text>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.</text>
```

5.9.6 RDF: Property rdfs:comment in class spdx:Snippet

Example:

```
<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>
```

5.10 Snippet Name

5.10.1 Purpose: Identify a specific snippet in a human convenient manner.

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

5.10.3 Cardinality: Optional, one.

5.10.4 Data Format: Single line of text

5.10.5 Tag: SnippetName:

Example:

```
SnippetName: from linux kernel
```

5.10.6 RDF: Property spdx:snippetName in class spdx:Snippet

Example:

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

6 Other Licensing Information Detected

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: http://spdx.org/licenses/. The SPDX License List can also be found here in Appendix I.

One instance should be created for every unique license or licensing information reference detected in package that does not match one of the licenses on the SPDX License List. Each license instance should have the following fields.

Fields:

6.1 License Identifier

6.1.1 Purpose: 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 and files sections of the SPDX file (sections 3 and 4, respectively).

6.1.2 Intent: Create a human readable short form license identifier for a license not on the SPDX License List. This identifier should be unique within the SPDX file. 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.

6.1.3 Cardinality: Conditional (mandatory, one) if license is not on SPDX License List.

6.1.4 Data Format: "LicenseRef-"[idstring]

where

[idstring] is a unique string containing letters, numbers, . and/or –.

6.1.5 Tag: LicenseID:

Examples:

```
LicenseID: LicenseRef-1

LicenseID: LicenseRef-Beerware-4.2
```

6.1.6 RDF: Property spdx:licenseID in class spdx:ExtractedLicensingInfo

Examples:

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

```

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

6.2 Extracted Text

6.2.1 Purpose: Provide a copy of the actual text of the license reference extracted from the package or file that is associated with the License Identifier to aid in future analysis.

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

6.2.3 Cardinality: Conditional (Mandatory, one) if there is a License Identifier assigned.

6.2.4 Data Format: Free form text field that may span multiple lines.

6.2.5 Tag: ExtractedText:

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

Example 1 (if only short reference to license present in File):

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

Example 2 (if indeed full text of license present in File):

```
ExtractedText: <text>"THE WHISKEY-WARE LICENSE": whiskeyfan@example.com wrote 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 worth it, you can buy me a bottle of whiskey in return </text>
```

6.2.6 RDF: Property spdx:extractedText in class spdx: ExtractedLicensingInfo

Example 1 (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 LICENSE.</extractedText>
</ExtractedLicensingInfo>
```

Example 2 (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 wrote 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 worth it, you can buy me a bottle of whiskey in return.</extractedText>
</ExtractedLicensingInfo>
```

6.3 License Name

6.3.1 Purpose: 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.

6.3.2 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 data to humans.

6.3.3 Cardinality: Conditional (mandatory, one) if license is not on SPDX License List.

6.3.4 Data Format: Single line of text | NOASSERTION.

6.3.5 Tag: LicenseName:

Example:

```
LicenseName: Whiskey—Ware License
```

6.3.6 RDF: Property spdx:licenseName in class spdx: ExtractedLicensingInfo

Example:

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

6.4 License Cross Reference

6.4.1 Purpose: 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.

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

6.4.3 Cardinality: Conditional (optional, one or more) if license is not on SPDX License List.

6.4.4 Data Format: Uniform Resource Locator

6.4.5 Tag: LicenseCrossReference:

Example:

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

6.4.6 RDF: Property rdfs:seeAlso in class spdx: ExtractedLicensingInfo

Example:

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

6.5 License Comment

6.5.1 Purpose: This field provides a place for the SPDX file creator to record any general comments about the license.

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

6.5.3 Cardinality: Optional, one.

6.5.4 Data Format: Free form text that can span multiple lines

6.5.5 Tag: LicenseComment:

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

Example:

```
LicenseComment: <text>The Whiskey—Ware License has a couple of other standard variants .</ text>
```

6.5.6 RDF: Property rdfs:comment in class spdx: ExtractedLicensingInfo

Example:

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

7 Relationships between SPDX Elements

7.1 Relationship

7.1.1 Purpose: 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. The relationships between two elements that are supported are:

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.
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
GENERATES	Is to be used when SPDXRef-A generates the SPDXRef-B.	A SOURCE file makefile.mk generates a BINARY file a.out

Relationship	Description	Example
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).

Relationship	Description	Example
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”.
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 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 was deleted from to 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.

Relationship	Description	Example
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 to build SPDXRef-B.	A SOURCE file makefile.mk is used to build an APPLICATION “zlib”.

Relationship	Description	Example
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.

Relationship	Description	Example
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 aprerequisite 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 of 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 should be included in the Relationship comments field.	

7.1.2 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.

7.1.3 Cardinality: Optional*, multiple.

* see DESCRIBES relationship for one mandatory case.

7.1.4 Data Format:

```
1  ["DocumentRef-"[idstring]":"]SPDXID <relationship> ["DocumentRef-"[idstring]":"]SPDXID
```

where "DocumentRef-"[idstring]":" is an optional reference to an external SPDX document as described in section 2.6

where SPDXID is a string containing letters, numbers, . and/or -. as described in sections (2.3, 3.2, 4.2).

where <relationship> is one of the documented relationship types in table 7.1.1.

7.1.5 Tag: Relationship :

Examples:

```
1  Relationship : SPDXRef-grep CONTAINS SPDXRef-make
2
3  RelationshipComment: Package grep contains file make
4
5  Relationship : SPDXRef-DOCUMENT AMENDS DocumentRef-SPDXA:SPDXRef-DOCUMENT
6
7  RelationshipComment: This current document is an amendment of the SPDXA document.
```

7.1.6 RDF: Property relationship in any SpdxElement

Examples:

```
1  <SpdxElement rdf:about"=#SPDXRef"-45>
2      <relationship>
3          <Relationship>
4              <spdx:relatedSpdxElement>
5                  <spdx:SpdxElement rdf:about="http://spdx.org/spdxdocs/spdx-tools-v1.2-3F2504E0-4F89
       -41D3-9A0C-0305E82...
6                  </spdx:relatedSpdxElement>
7
8                  <relationshipType>http :// spdx.org/ rdf /terms#relationshipType_contains</ relationshipType>
9              </ Relationship>
10         </ relationship >
11
12     ...
13
14 </SpdxElement>
```

7.2 Relationship Comment

7.2.1 Purpose: This field provides a place for the SPDX file creator to record any general comments about the relationship.

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

7.2.3 Cardinality: Optional, one.

7.2.4 Data Format: Free form text that can span multiple lines, refers only to the immediately preceding relationship.

7.2.5 Tag: RelationshipComment:

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

A RelationshipComment: must be the line immediately after a "Relationship:"

Example:

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

7.2.6 RDF: Property rdfs:comment in class spdx:Relationship

Example:

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

     ...

</Relationship>
```

8 Annotations

8.1 Annotator

8.1.1 Purpose: This field identifies the person, organization or tool that has commented on a file, package, or the entire document.

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

8.1.3 Cardinality: Conditional (Mandatory, one), if there is an Annotation.

8.1.4 Data Format: Single line of text with the following keywords.

```
1  ”
2  Person: person ”name and optional “(email)”
3  ”Organization: ”organization and optional “(email)”
4  ”Tool: tool identifier — ”version
```

8.1.5 Tag: Annotator:

Example:

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

8.1.6 RDF: Property spdx:annotator in class spdx:Annotation

Example:

```
1  <Annotation>
2      <annotator> Person: Jane Doe () </annotator>
3  </Annotations>
```

8.2 Annotation Date

8.2.1 Purpose: Identify when the comment was made. This is to be specified according to the combined date and time in the UTC format, as specified in the ISO 8601 standard.

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

8.2.3 Cardinality: Conditional (Mandatory, one), if there is an Annotation.

8.2.4 Data 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

8.2.5 Tag: AnnotationDate:

Example:

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

8.2.6 RDF: Property spdx:annotationDate in class spdx:Annotation

Example:

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

8.3 Annotation Type

8.3.1 Purpose: This field describes the type of annotation. Annotations are usually created when someone reviews the file, 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.

8.3.2 Intent: This allows the type of annotation to be recorded.

8.3.3 Cardinality: Conditional (Mandatory, one), if there is an Annotation.

8.3.4 Data Format: REVIEW | OTHER

8.3.5 Tag: AnnotationType:

Example:

```
AnnotationType: REVIEW
```

8.3.6 RDF: property annotationType in class spdx:Annotation

Example:

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

8.4 SPDX Identifier Reference

8.4.1 Purpose: 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.

8.4.2 Intent: There may be several versions of the same 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.

8.4.3 Cardinality: Conditional (Mandatory, one), if there is an Annotation.

8.4.4 DataFormat: [DocumentRef—[idstring]:]SPDXID

where:

["DocumentRef-"[idstring]":"] is an optional reference to an external SPDX document as described in section 2.6 SPDXID is a unique string containing letters, numbers, . and/or — as described in Sections 2.3, 3.2 and 4.2.

8.4.5 Tag: SPDXREF:

Example:

```
SPDXREF: SPDXRef—45
```

Example:

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

8.4.6 RDF:

For RDF, the annotations are a property of the SPDX element it is annotationg.

```
<SpdxElement rdf:about"=#SPDXRef"—45>
    <annotation>
        <Annotation>
            ...
        </Annotation>
    </annotation>
</SpdxElement rdf:about"=#SPDXRef"—45>
```

8.5 Annotation Comment

8.5.1 Purpose: This optional free form text field permits the annotator to provide commentary on the analysis.

8.5.2 Intent: This allows the annonator to provide independent assessment and note any points where there is disagreement with the analysis.

8.5.3 Cardinality: Conditional (Mandatory, one), if there is an Annotation.

8.5.4 Data Format: Free form text that can span multiple lines.

8.5.5 Tag: AnnotationComment:

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

Example:

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

8.5.6 RDF: Property rdfs:comment in class spdx:Annotation

Example:

```
<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 influence the concluded license, and some alternatives may be possible,
    but the concluded license is one of the options.
    </rdfs:comment>
</Annotation>
```

9 Review Information (deprecated)

The review information section is included for compatibility with SPDX 1.2, and is deprecated since SPDX 2.0. Any review information should use an Annotation (as described in section 8) with an annotation type of annotationType_review.

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

Fields:

9.1 Reviewer (deprecated)

This field has been deprecated since SPDX 2.0.

9.1.1 Purpose: This field identifies the person, organization or tool that has reviewed the SPDX file. This field is optional and thus there is no requirement for any reviewer to add a set of review information to the file. 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 file is created and be appended to the original file.

9.1.2 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 file.

9.1.3 Cardinality: Optional, one.

9.1.4 Data Format: Single line of text with the following keywords.

```
1 ”
2 Person: person ”name and optional “(email)”
3 ”Organization: ”organization and optional “(email)”
4 ”Tool: tool identifier — ”version
```

9.1.5 Tag: Reviewer:

Example:

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

9.1.6 RDF: Property spdx:reviewer in class spdx:Review

Example:

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

9.2 Review Date (deprecated)

This field has been deprecated since SPDX 2.0.

9.2.1 Purpose: Identify when the review was done. This is to be specified according to the combined date and time in the UTC format, as specified in the ISO 8601 standard.

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

9.2.3 Cardinality: Conditional (Mandatory, one), if there is a Reviewer.

9.2.4 Data 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

9.2.5 Tag: ReviewDate:

Example:

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

9.2.6 RDF: Property spdx:reviewDate in class spdx:Review

Example:

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

9.3 Review Comment (deprecated)

This field is deprecated since SPDX 2.0.

9.3.1 Purpose: This optional free form text field permits the reviewer to provide commentary on the analysis.

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

9.3.3 Cardinality: Optional, one.

9.3.4 Data Format: Free form text that can span multiple lines.

9.3.5 Tag: ReviewComment:

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

Example:

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

9.3.6 RDF: Property rdfs:comment in class spdx:Review

Example:

```
<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 influence the concluded license, and some alternatives may be possible,
    but the concluded license is one of the options.</rdfs:comment>
</Review>
```

Appendix I: 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.
- Master Files: The HTML pages you see here are generated from the master files for the SPDX License List. The master files include 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.
- 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, v2.5 which was released July 2016. 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/

I.1 Licenses with 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	
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	AGPL-1.0	
GNU Affero General Public License v3.0	AGPL-3.0	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

Full Name of License	Short Identifier	OSI?
Apple Public Source License 2.0	APSL-2.0	Y
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	
Borceux license	Borceux	
“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 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	
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 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	

Full Name of License	Short Identifier	OSI?
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	CC-BY-1.0	
Creative Commons Attribution 2.0	CC-BY-2.0	
Creative Commons Attribution 2.5	CC-BY-2.5	
Creative Commons Attribution 3.0	CC-BY-3.0	
Creative Commons Attribution 4.0	CC-BY-4.0	
Creative Commons Attribution Non Commercial 1.0	CC-BY-NC-1.0	
Creative Commons Attribution Non Commercial 2.0	CC-BY-NC-2.0	
Creative Commons Attribution Non Commercial 2.5	CC-BY-NC-2.5	
Creative Commons Attribution Non Commercial 3.0	CC-BY-NC-3.0	
Creative Commons Attribution Non Commercial 4.0	CC-BY-NC-4.0	
Creative Commons Attribution Non Commercial No Derivatives 1.0	CC-BY-NC-ND-1.0	
Creative Commons Attribution Non Commercial No Derivatives 2.0	CC-BY-NC-ND-2.0	
Creative Commons Attribution Non Commercial No Derivatives 2.5	CC-BY-NC-ND-2.5	
Creative Commons Attribution Non Commercial No Derivatives 3.0	CC-BY-NC-ND-3.0	
Creative Commons Attribution Non Commercial No Derivatives 4.0	CC-BY-NC-ND-4.0	
Creative Commons Attribution Non Commercial Share Alike 1.0	CC-BY-NC-SA-1.0	
Creative Commons Attribution Non Commercial Share Alike 2.0	CC-BY-NC-SA-2.0	

Full Name of License	Short Identifier	OSI?
Creative Commons Attribution Non Commercial Share Alike 2.5	CC-BY-NC-SA-2.5	
Creative Commons Attribution Non Commercial Share Alike 3.0	CC-BY-NC-SA-3.0	
Creative Commons Attribution Non Commercial Share Alike 4.0	CC-BY-NC-SA-4.0	
Creative Commons Attribution No Derivatives 1.0	CC-BY-ND-1.0	
Creative Commons Attribution No Derivatives 2.0	CC-BY-ND-2.0	
Creative Commons Attribution No Derivatives 2.5	CC-BY-ND-2.5	
Creative Commons Attribution No Derivatives 3.0	CC-BY-ND-3.0	
Creative Commons Attribution No Derivatives 4.0	CC-BY-ND-4.0	
Creative Commons Attribution Share Alike 1.0	CC-BY-SA-1.0	
Creative Commons Attribution Share Alike 2.0	CC-BY-SA-2.0	
Creative Commons Attribution Share Alike 2.5	CC-BY-SA-2.5	
Creative Commons Attribution Share Alike 3.0	CC-BY-SA-3.0	
Creative Commons Attribution Share Alike 4.0	CC-BY-SA-4.0	
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	
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	
Clarified Artistic License	ClArtistic	
CNRI Jython License	CNRI-Jython	
CNRI Python License	CNRI-Python	Y

Full Name of License	Short Identifier	OSI?
CNRI Python Open Source GPL Compatible License Agreement	CNRI-Python-GPL-Compatible	
Condor Public License v1.1	Condor-1.1	
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	
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
Erlang Public License v1.1	ErlPL-1.1	
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

Full Name of License	Short Identifier	OSI?
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	
Freetype Project License	FTL	
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	
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	
GNU General Public License v2.0 only	GPL-2.0	Y
GNU General Public License v3.0 only	GPL-3.0	Y
gSOAP Public License v1.3b	gSOAP-1.3b	
Haskell Language Report License	HaskellReport	
Historic Permission Notice and Disclaimer	HPND	Y
IBM PowerPC Initialization and Boot Software	IBM-pibs	
ICU License	ICU	
Independent JPEG Group License	IJG	
ImageMagick License	ImageMagick	
iMatix Standard Function Library Agreement	iMatix	

Full Name of License	Short Identifier	OSI?
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	
JSON License	JSON	
License Art Libre 1.2	LAL-1.2	
License Art Libre 1.3	LAL-1.3	
Latex2e License	Latex2e	
Leptonica License	Leptonica	
GNU Library General Public License v2 only	LGPL-2.0	Y
GNU Lesser General Public License v2.1 only	LGPL-2.1	Y
GNU Lesser General Public License v3.0 only	LGPL-3.0	Y
Lesser General Public Licenses For Linguistic Resources	LGPLLR	
libpng License	Libpng	
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
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	

Full Name of License	Short Identifier	OSI?
LaTeX Project Public License v1.1	LPPL-1.1	
LaTeX Project Public License v1.2	LPPL-1.2	
LaTeX Project Public License 1.3a	LPPL-1.3a	
LaTeX Project Public License v1.3c	LPPL-1.3c	Y
MakeIndex License	MakeIndex	
MirOS Licence	MirOS	
MIT License	MIT	Y
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	
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
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	
Multics License	Multics	
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

Full Name of License	Short Identifier	OSI?
NetCDF license	NetCDF	
Newsletr License	Newsletr	
Nethack General Public License	NGPL	Y
Norwgian License 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	Y
Non-Profit Open Software License 3.0	NPOSL-3.0	Y
NRL License	NRL	
NTP License	NTP	Y
Nunit License	Nunit	
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	
SIL Open Font License 1.0	OFL-1.0	
SIL Open Font License 1.1	OFL-1.1	Y
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	

Full Name of License	Short Identifier	OSI?
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	
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
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
psfrag License	psfrag	
psutils License	psutils	
Python License 2.0	Python-2.0	Y
Qhull License	Qhull	

Full Name of License	Short Identifier	OSI?
Q Public License 1.0	QPL-1.0	Y
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	
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	
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	
Spencer License 94	Spencer-94	
Spencer License 99	Spencer-99	
Sun Public License v1.0	SPL-1.0	Y

Full Name of License	Short Identifier	OSI?
SugarCRM Public License v1.1.3	SugarCRM-1.1.3	
Scheme Widget Library (SWL) Software License Agreement	SWL	
TCL/TK License	TCL	
TMate Open Source License	TMate	
TORQUE v2.5+ Software License v1.1	TORQUE-1.1	
Trusster Open Source License	TOSL	
Unicode Terms of Use	Unicode-TOU	
The Unlicense	Unlicense	
Universal Permissive Licenses 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
W3C Software Notice and License (2002-12-31)	W3C	Y
W3C Software Notice and License (1998-07-20)	W3C-19980720	
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	

Full Name of License	Short Identifier	OSI?
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	

I.2 Exceptions List

Full Name of Exception	SPDX LicenseException
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
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

Full Name of Exception	SPDX LicenseException
i2p GPL+Java Exception	i2p-gpl-java-exception
Libtool Exception	Libtool-exception
LZMA exception	LZMA-exception
Macros and Inline Functions Exception	mif-exception
Nokia Qt LGPL exception 1.1	Nokia-Qt-exception-1.1
Open CASCADE Exception 1.0	OCCT-exception-1.0
OpenVPN OpenSSL Exception	openvpn-openssl-exception
Qwt exception 1.0	Qwt-exception-1.0
U-Boot exception 2.0	u-boot-exception-2.0
WxWindows Library Exception 3.1	WxWindows-exception-3.1

I.3 Deprecated Licenses

Full Name of License	Deprecated SPDX License Identifier
eCos license version 2.0	eCos-2.0
GNU General Public License v1.0 or later	GPL-1.0+
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
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 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 Lesser General Public License v2.1 or later	LGPL-2.1+

Full Name of License	Deprecated SPDX License Identifier
GNU Lesser General Public License v3.0 or later	LGPL-3.0+
GNU Library General Public License v2 or later	LGPL-2.0+
Standard ML of New Jersey License	StandardML-NJ
wxWindows Library License	WXwindows

Appendix II: License Matching Guidelines and Templates

The SPDX License List Matching Guidelines provide guidelines to be used for the purposes of matching licenses and license exceptions against those found on the SPDX License List. There is no intent here to make a judgment or interpretation, but merely to ensure that when one SPDX creator 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 with markups.

SPDX License List 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. 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.

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 omitable as per Matching Guideline #2 (“Substantive Text”).
  - Indicated by «var; . . . » or…
  - Indicated by «beginOptional; . . .» and <> 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:

```
1    Semicolons are escaped with \;
2
3    POSIX Bracket Extensions are not allowed
```

Example:

```
1  <<var;name=organizationClause3;original=the copyright holder;match=.+>>
```

Appendix III: RDF Data Model Implementation and Identifier Syntax

SPDX ® Vocabulary Specification

See: http://spdx.org/rdf/ontology/spdx-2-1

Version: 2.1

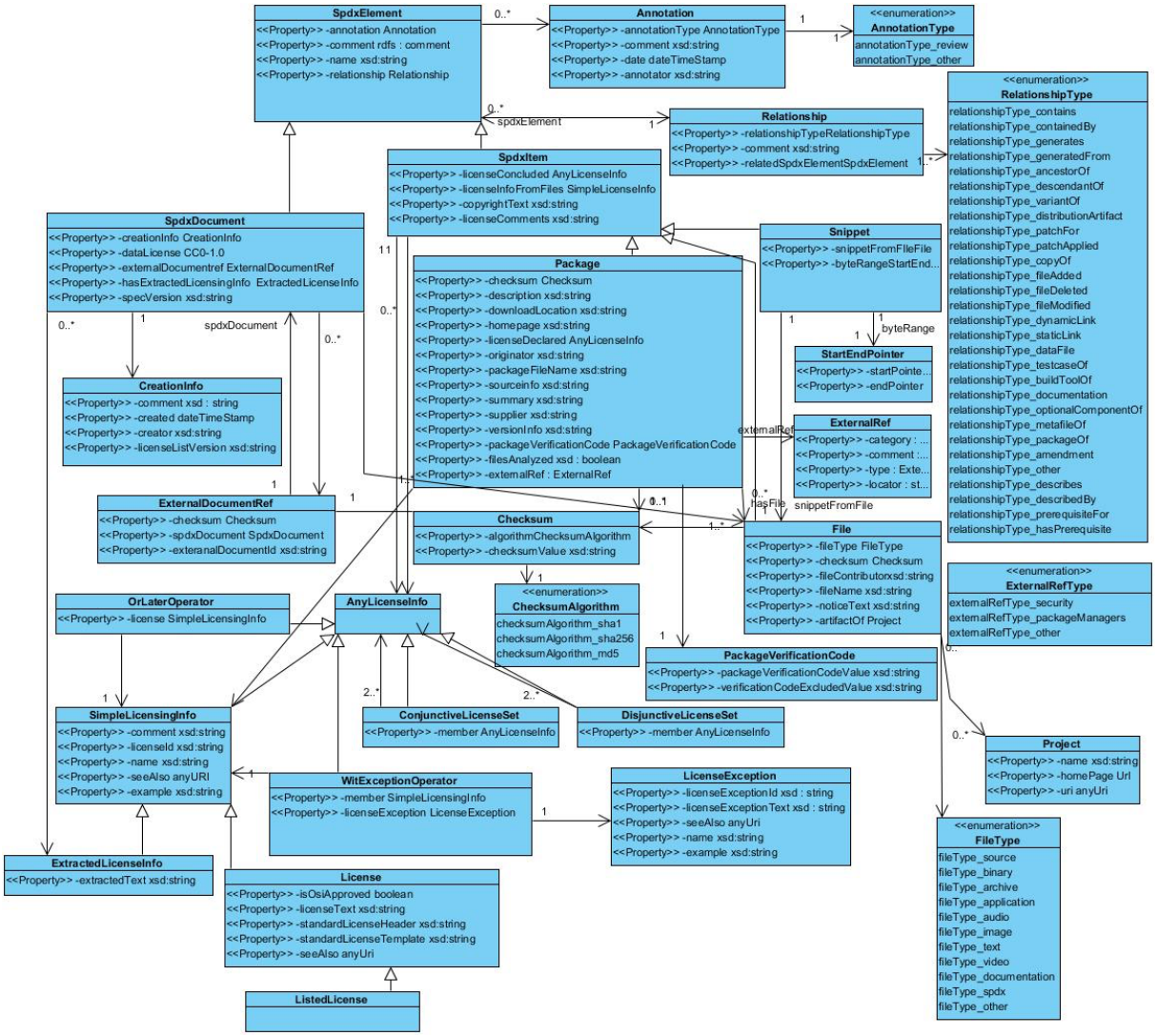

Figure 2: SPDX 2.1 RD Ontology

Agent and Tool Identifiers

Fields that identify entities that have acted in relation to the SPDX file 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 “ool: LicenseFind - 1.0”. The exact syntax of agent and tool identifications is described below in ABNF.

```
1  agent            = person / organization
2
```

```
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+0022

contact-info        = " (" email-addr ")"

email-addr          = local-name-atom *( "." local-name-atom ) "@" domain-name-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
```

Appendix IV: SPDX License Expressions

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 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 AND BSD-3-Clause).

SPDX License Expressions provides 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 Appendix I.1>

license-exception-id = <short form license exception identifier in Appendix I.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 whitespace between a license-id and any following +. This supports easy parsing and backwards compatibility. There MUST be whitespace on either side of the operator "WITH". There MUST be whitespace and/or parentheses on either side of the operators AND and OR.

Simple License Expressions

A simple <license—expression> is composed one of the following:

- An SPDX License List Short Form Identifier. For example: GPL-2.0
- 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: GPL-2.0+
- A 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
```

Composite License Expressions

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, should be encapsulated by parentheses: "( )". This has been specified to facilitate expression parsing. Nested parentheses can also be used to specify an order of precedence which is discussed in more detail in subsection (4).

1) Disjunctive "OR" Operator

If presented with a choice between two or more licenses, use the disjunctive binary "OR" operator to construct a new lincense expression, where both the left and right operands are valid license expression values.

For example, when given a choice between the LGPL-2.1 or MIT licenses, a valid expression would be:

```
(LGPL-2.1 OR MIT)
```

An example representing a choice between three different licenses would be:

(LGPL-2.1 OR MIT OR BSD-3-Clause)

2) Conjunctive "AND" Operator

If required to simultaneously comply with two or more licenses, use the conjunctive binary "AND" operator to construct a new license expression, where both the left and right operands are a valid license expression values.

For example, when one is required to comply with both the LGPL-2.1 or MIT licenses, a valid expression would be:

```
(LGPL-2.1 AND MIT)
```

An example where all three different licenses apply would be:

```
(LGPL-2.1 AND MIT AND BSD-2-Clause)
```

3) 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+, the expression would be:

```
(GPL-2.0+ WITH Bison-exception-2.2)
```

The current set of valid exceptions can be found in Appendix I, section 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).

4) 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 OR BSD-3-Clause AND MIT
```

represents a license choice between either LGPL-2.1 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 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 the BSD-3-Clause license before applying the MIT license.

5) 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"/>
    <spdx:ExtractedLicensingInfo rdf:about="http://example.org#LicenseRef-EternalSurrender">
        <spdx:extractedText>
            In exchange for using this software, you agree to give its author all your worldly possessions.
            You will not hold the author liable for all the damage this software will inevitably cause not only
            to your person and property, but to the entire fabric of the cosmos.
        </spdx:extractedText>
        <spdx:licenseId>LicenseRef-EternalSurrender</spdx:licenseId>
    </spdx:ExtractedLicensingInfo>
</spdx:ConjunctiveLicenseSet>
```

A disjunctive license can be expressed in RDF via a <spdx: DisjunctiveLicenseSet > element, with an spdx:member property for each element in the disjunctive license. Two or more members are required.

```
<spdx: DisjunctiveLicenseSet >
    <spdx:member rdf:resource="http://spdx.org/licenses/GPL-2.0"/>
    <spdx:member>
        <spdx: ExtractedLicensingInfo rdf:about="http://example.org#LicenseRef-EternalSurrender">
            <spdx:extractedText>
                In exchange for using this software, you agree to give its author all your worldly possessions.
                You will not hold the author liable for all the damage this software will inevitably cause not only to your person and property, but to the entire fabric of the cosmos.
            </spdx:extractedText>
            <spdx:licenseId>LicenseRef-EternalSurrender</spdx:licenseId>
        </spdx: ExtractedLicensingInfo >
    </spdx:member>
</spdx: DisjunctiveLicenseSet >
```

A License Exception can be expressed in RDF via a <spdx:LicenseException> element. This element has the following attributes:

- Comment - An rdfs:comment element describing the nature of the exception.
- See Also (optional)- An rdfs:seeAlso element referencing external sources of information on the exception.
- Example - Text describing examples of this exception.
- Name - The full human readable name of the item.
- License Exception ID: The identifier of an exception in the SPDX License List to which the exception applies.
- License Exception Text: Full text of the license exception.

```
<rdf:Description rdf:about="http://example.org#SPDXRef-ButIdDontWantToException">
    <rdfs:comment>This exception may be invalid in some jurisdictions .</rdfs:comment>
    <rdfs:seeAlso>http://dilbert.com/strip/1997-01-15</rdfs:seeAlso>
    <spdx:example>So this one time, I had a license ...exception</spdx:example>
    <spdx:licenseExceptionText>
        A user of this software may decline to follow any subset of the terms of this license upon
        finding any or all such terms unfavorable.
    </spdx:licenseExceptionText>
    <spdx:name>&quot;But I Don&apos;t Want To&quot; Exception</spdx:name>
    <spdx:licenseExceptionId>SPDXRef-ButIdDontWantToException</spdx:licenseExceptionId>
    <rdf:type rdf:resource="http://spdx.org/rdf/terms#LicenseException"/>
```

```
12      </rdf:Description>
```

Appendix V: Using SPDX short identifiers in Source Files

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 lookups 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 the examples on the SPDX WIKI page about Meta_Tags.

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:

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: GPL—2.0+
SPDX—License—Identifier: MIT
```

Representing Multiple Licenses

Multiple licenses can be represented using a SPDX license expression as defined in Appendix IV. A set of licenses must be enclosed in parentheses (this is a convention for SPDX expressions). As further described there:

1. When there is a choice between licenses ("disjunctive license"), they should be separated with "OR". If presented with a choice between two or more licenses, use the disjunctive binary "OR" operator to construct a new license expression.
2. Similarly when multiple licenses need to be simultaneously applied ("conjunctive license"), they should be separated with "AND". If required to simultaneously comply with two or more licenses, use the conjunctive binary "AND" operator to construct a new license expression.
3. In some cases, a set of license terms apply except under special circumstances, in this case, use the "WITH" operator followed by one of the recognized exception identifiers.
4. 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.

Examples:

```
SPDX—License—Identifier: (GPL—2.0 OR MIT)
SPDX—License—Identifier: (LGPL—2.1 AND BSD—2—CLAUSE)
SPDX—License—Identifier: (GPL—2.0+ WITH Bison—exception—2.2)
```

Please see Appendix IV of SPDX 2.1 Specification 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: http://spdx.org/spdx-license-list/request-new-license-or-exception.

Appendix VI: External Repository Identifiers

When <category> = SECURITY: ***

<type> **cpe22Type**

<locator> **Information**

Locator Format:

```
1  "[c][pP][eE]:/[AHOaho]?(:[A-Za-z0-9\._\-~%]*){0,6}"
```

Contextual Example:

```
1  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

<type> **cpe23Type**

<locator> **Information**

Locator Format:

```
1  "cpe:2\.3:[aho\*\]
2    (:((((\?*|\*?)  ([  azAZ09  \  \._  ]|(\\[\\\*\?!
3  "#$$%&'\(\) \+,/:;<=> @  \[\]\^`\{\|}~])
4  ) +(\?*|\*?) ) |[\*\  ]) ) {5}
```

```
  (:(([  azAZ  ]{2,3}(  ([  azAZ  ]{2}|[0  9 ]{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

---

When = PACKAGE_MANAGER: ***

<type> **maven-central**

<locator> **Information**

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

<type> **npm**

<locator> **Information**

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

<type> nuget

<locator> Information

Locator Format:

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

Contextual Example:

```
Microsoft .AspNet.MVC/5.0.0
```

External Reference Site: https://www.nuget.org/

Documentation: https://docs.nuget.org/

<type> bower

<locator> Information

Locator Format:

```
    package#version
    ^[^#]+#[^#]+$
```

Contextual Example:

```
    modernizr#2.6.2
```

External Reference Site: http://bower.io/

Documentation: http://bower.io/docs/api/#install

When = OTHER: ***

<type> **[idstring]**

<locator> **Information**

No spaces, but anything else goes

Appendix VII: Creative Commons Attribution License 3.0 Unported

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

a. **"Adaptation"** means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. **"Collection"** means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c. **“Distribute”** means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

d. **“Licensor”** means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

e. **“Original Author”** means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

f. **“Work”** means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. **“You”** means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

h. **“Publicly Perform”** means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. **“Reproduce”** means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work,

including storage of a protected performance or phonogram in digital form or other electronic medium.

2. **Fair Dealing Rights**. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.
3. **License Grant**. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;

b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";

c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,

d. to Distribute and Publicly Perform Adaptations.

e. For the avoidance of doubt:

f. **Non-waivable Compulsory License Schemes**. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

ii. **Waivable Compulsory License Schemes**. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

iii. **Voluntary License Schemes**. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. **Restrictions**. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.

b. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv) , consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4 (b) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author,

Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

c. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. **Representations, Warranties and Disclaimer**

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. **Limitation on Liability**. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. **Termination**

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any

time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. **Miscellaneous**

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.