How-To Guide for SBOM Generation

Table of Contents

1 Introduction

A *software bill of materials* (SBOM) is a structured list of underlying components that are included in a piece of software. The SBOM captures the hierarchical relationships between the components. Because each of these components has potential security implications independent of the software in which it is included, awareness of these sometimes-hidden components is critical to understanding the current and future cybersecurity vulnerabilities of any software.

While there is no standard, regulation or industry guidance for creation and maintenance of SBOMs, the Healthcare Proof-of-Concept working group coordinated by the US NTIA Software Component Transparency group collected simple instructions and guidance on how to generate an SBOM. The information provided is based on the knowledge acquired through the generation of SBOMs for the POC. Although developed as part of the Healthcare POC, it is expected that this guidance can be applied to the creation and maintenance of SBOMs in any industry.

2 SBOM Basics

An SBOM begins when a development team selects a software component for inclusion in their code and builds. These components can be publicly available open source software (OSS), purchased commercial off the shelf (COTS) libraries from a 3rd party, internal shared common code modules, etc. Each of the included components is one element of the SBOM for the software being developed. The collection of all the elements makes up the full SBOM.

The software development team is ideally positioned to be the creator and maintainer of the SBOM for their software. Modern engineering practice dictates that engineering teams maintain and understand the modules used in their code. Many software development tools provide automated SBOM creation and maintenance.

Once the software for a product is released, by definition the SBOM content for that product is also considered complete. An SBOM is associated with a single release of software, and each new released version of code requires a new SBOM.

At a minimum, the software identity of a component can be established by a supplier name, component name and version string. A code hash may be used to further strengthen unique identification of the component, although the Healthcare POC has not yet evaluated the use of a code hash.

The development and build teams should maintain the SBOM as part of the regular build and release process for the software it describes, and should make it available to consumers of the software using some agreed-upon method.

3 Collecting SBOM Content

Once SBOMs are part of product security requirements and the standard software engineering build and release process, creating and communicating the SBOM becomes a by-product of all releases. However, for organizations who do not yet have – or who are in the process of establishing - SBOM processes, it is likely that engineering teams are not yet maintaining or providing the list of included components. If that's the case, there are some alternate routes to collecting the SBOM information.

3.1 Product and Engineering Team Resources

The first stop for any fundamental information about a code base should be with the management of the responsible development teams. In many cases, the engineering managers can provide an existing list of components that is already being maintained or easily created in response to a request.

Another possible source is the manager of product requirements or product manager. In some organizations, the product manager's consent is needed for any action by engineering. Also, the requirements manager often specifies security requirements for the software, and if so a specific requirement for SBOM should be considered. But regardless of the organization and approach, the person who defines requirements can act as a bridge between engineering and the security and compliance teams to help obtain SBOMs.

Finally, product development documentation such as System Design Documents (SDD), Detail Design Documents (DDD), Version Description Documents (VDD), and build scripts often provide granular information about the composition of the software. For example, OSS (Open Source Software) projects on GitHub often include a README with information about the software that can include package/module/library/widget dependency information, and Nodejs automatically creates a package.json with a listing of the modules in use, with version numbers.

3.2 Software Composition Analysis (SCA) and Binary Code Analysis Tools

If the engineering or product management team is unavailable to help, there are software analysis tools available to scan the software to identify the included components.

Software composition analysis (SCA) and binary composition analysis tools have several advantages. The biggest is that they can be used to audit software without any input from engineering team (or, in the case of a consumer, from the supplier). Many will generate SBOMs in standard formats and automatically feed configuration and asset databases. Finally, many tools cross reference a variety of defect and security databases to correlate known vulnerabilities with components.

There are also significant disadvantages. All the tools rely on proprietary databases and code fingerprints which may be incomplete or out of date, and therefore may not properly identify every included component. A limitation of source code tools is that software source code files may not be available. Another disadvantage is that SCA may produce false positives, identifying software components or dependencies that were not included or used in the development of the software. Perhaps the biggest disadvantage is that the SBOM program will not be supported as a part of development and may therefore not have definitive information.

3.3 Legal/Licensing/Compliance

Many organizations maintain license databases or even paper files to track and comply with license requirements. Licenses are a source of great legal, financial, and intellectual property (IP) risk – and many companies track them thoroughly to ensure compliance. The licenses for all components contained in each piece of software should be provided to the organization using that software by the supplier, and many companies use tools like CMDB and asset managers. If such a database is well-maintained, it can be a reasonable source for SBOM data. However, because there can be gaps in license information, this method is probably best used as a secondary source for verification.

3.4 Baseline Elements Required for an SBOM

The Framing Working Group of the NTIA Software Component Transparency initiative published the document *Framing Software Component Transparency: Establishing a Common Software Bill of Material (SBOM)* to establish the baseline content for an SBOM. The following baseline elements were included as part of the Healthcare POC, and are covered by this guide (framing chapter numbers are prefixed with baseline element names for ease of identification):

- 2.2.1 **Author Name** – the author of the SBOM document describing the Primary Component. The author may not be the same as the supplier of the Primary Component.
- 2.2.2 **Supplier Name** – the supplier of a component.
- 2.2.3 **Component Name** – the name of the component.
- 2.2.4 **Version String** – the version of the component.
- 2.2.5 **Component Hash** –a cryptographic hash used to identify the binary instance of a component. It has not been investigated yet by the Healthcare POC, and is not currently provided in the SBOM content.
- 2.2.6 **Unique Identifier** – a unique identifier for a component. Multiple identifiers may exist for a component because different systems may use a different identifier.
- 2.2.7 **Relationship** – used to establish that a component is included in another component. In addition, Relationship is used to document knowledge about the completeness of the list of components included in another component.
- 2.4 Component Relationships
    - **Primary Component** – the component described by the SBOM
    - **Included Component** – the components included in another component

Please refer to the Framing document for additional details on these elements.

4 Generation Guidance

The section provides guidance on important SBOM concepts to clarify how to establish the content needed to generate the SBOM.

4.1 Software Identity

The component's **Software Identity** is standardized for the Healthcare POC as the **Supplier Name**, **Component Name**, and **Version String** baseline elements. For the **Primary Component**, the SBOM author is responsible for defining these values. For the **Included Components**, the following approach is recommended:

- Obtain these values for the SBOM published for the component, otherwise
- Obtain these values from the supplier, otherwise
- Obtain these values from an existing system, or
  - For example, use the CPE information from the NVD maintained by NIST
- Establish with a best-effort approach to define the values

One of the current challenges with SBOM generation is creating consistent and unique Software Identities for the same component across SBOMs. The guidance above was followed to obtain a reasonable level of consistency and uniqueness.

4.2 Unique Identifier

This should be a globally unique value that refers to a component. The Healthcare POC explored various options and decided to establish a supplier-provided unique identifier using a package URL (*purl*) that defines an URL string. Because the identifier is based on the **Software Identity** elements, the ***purl*** identifier will be unique if the **Software Identity** is unique.

The following ***purl*** conventions were used to construct a **Unique Identifier** for each SBOM component:

- Construct a ***purl*** identifier using **Supplier Name**, **Component Name**, and **Version String**
- Use the following **purl** syntax to construct the URL
  - scheme:type/namespace/name@version?qualifiers#subpath
    - scheme: (pkg)
    - type: (supplier)
    - namespace: **Supplier Name**
    - name: **Component Name**
    - version: **Version String**
    - qualifiers#subpath (optional, not used)
  - pkg:supplier/<**Supplier Name**>/<**Component Name**>@<**Version String**>

The use of supplier for the type establishes that the identifier is being constructed from the information available to the author for the SBOM generation. Therefore, the identifier should be considered as the **Unique Identifier** used internally by the supplier.

Other unique identifiers, such as a Common Platform Enumeration (CPE) from NIST, can also be provided in addition to the **purl** identifier. Also, by using a unique namespace for the document, relative identifiers within the document can be considered unique identifiers for the component too. SPDX uses this approach, and additional information can be found in Section 5 SPDX Format of this document.

4.3 SBOM Completeness

Ideally, every software manufacturer would produce SBOM for their software components. However, the SBOM concepts and information are still evolving and SBOMs are not available for many software components, nor is tooling always available to open source projects so that they can provide them. In case of a missing SBOM for upstream software, the downstream or end-product manufacturer must create SBOM content based on available information of the upstream software components. This may lead to incomplete information in the final or end product's SBOM. See section *2.4.1 Knowledge About Relationships* in the Framing Group document for more information.

To provide completeness information about the SBOM, the following guidance was used based on the relationship assertion categories from the Framing Group document.

4.3.1 Completeness assertion for the Primary Component

The assertion for the **Primary Component** applies to the knowledge about the components directly included. The categories established by the Framing Group document were used by the Healthcare POC as follows:

- Unknown
  - No attempt was been made to identify **Included Components**
- Partial
  - A best-effort approach was used to identify **Included Components**, or
  - A Software Composition Analysis (SCA) was performed to identify **Included Components**, and therefore the list of components might be incomplete
- Known
  - All **Included Components** were identified and are listed in the SBOM
- Root
  - The **Primary Component** contains no **Included Components**

4.3.2 Completeness assertion for each Included Component

A completeness assertion was provided for each **Included Component**. The assertion establishes the knowledge a supplier has about the component based on the availability of an SBOM for the **Included Component** or efforts made by the supplier to identify the components of the **Included Component**.

The Healthcare POC used the categories in the following manner:

- Unknown
    - An SBOM was not collected, and
    - A SCA was not performed
- Partial
    - An SBOM was obtained and the completeness for the **Primary Component** described by the SBOM is "Partial," or
    - An SCA was performed to identify the next-level **Included Components**
- Known
    - An SBOM was obtained and the completeness for the **Primary Component** described by the SBOM is "Known"
- Root
    - An SBOM was obtained and the completeness for the **Primary Component** described by the SBOM is "Root"

4.4 SBOM Document Identity

It is also important to identify the SBOM document because there could be more than one version of the SBOM document for the same version of a software. For example, a newer version of the SBOM may be created to correct erroneous information in an older version of the SBOM document. The latest version of the SBOM document is assumed to have the most accurate and complete information about the software.

The SBOM Document Identity concept is not covered in the Framing Group document, but rather was discovered through the Healthcare POC as additional content necessary for the SBOM. The SBOM Document Identity was composed of the following elements:

- **SBOM Author** (same as 2.2.1 **Author Name**)
- **SBOM Document Name**
- **SBOM Document Version**

Like the **Software Identity** elements, authors should use logical conventions to establish the **SBOM Document Name** and **SBOM Document Version**.

In addition, the Healthcare POC identified a relationship between the **Author Name** and **Software Identity** of the **Primary Component**. If the author of the SBOM is not the same as the **Supplier** of the **Primary Component**, then it is likely that knowledge about the **Primary Component** is incomplete.

Capturing the time that the SBOM document was created is a good practice for versioning the SBOM to aid in determining the most recent and authoritative SBOM for the software component.

4.5 SBOM Content for Included Components

Two options were considered by the Healthcare POC for providing the SBOM content for **Included Components**. The content could be contained in the end-product SBOM, or a reference to the location of the upstream SBOM could be provided. Guidance on the use of these approaches is provided below.

4.5.1 Reference to External SBOM Document

If an SBOM for an **Included Component** is available, then either a reference to the SBOM external document or a reference to an external document that contains a list of SBOMS for multiple **Included Components** can be used. The reference information required depends on the format being used to describe the SBOM content.

4.5.2 Include in the SBOM Document

In situations where the SBOM for **Included Component** does not exist, the next-level SBOM content can be directly provided in the end product's SBOM. This information could come from an SCA performed to identify the composition of the **Included Component**, or even build information when the composition of the **Included Component** is unique to this product (e.g., subset of distributed package).

4.6 SBOM for Medical Device System-of-Systems

Some medical devices can be a combination of several sub-systems. In this case, the main medical system acts as a container for the sub-systems. This section explains how the component information of a system-of-systems can be captured by an SBOM that could contain software and hardware components that contain software.

The following assumptions were made about the system-of-systems:

- One endpoint can have multiple configurations
    - An SBOM is required for each configuration
    - A unique Software Identity is required for each configuration
- The “sellable” unit should have an SBOM
- An SBOM may be provided for a sub-system

The Healthcare POC determined that the same conventions established for providing SBOM Content for **Included Components** can be applied in the system-of-systems scenario. The medical device is the **Primary Component**, and the sub-systems are **Included Components**. The SBOM author decides whether the next-level SBOM content will be directly included or if an external reference will be used.

The Healthcare POC defined two examples of a medical device system of systems. One example represents a product with a single endpoint, while the other example represents a product with multiple endpoints. A complete example of a product with single endpoint and the associated SBOMs is provided in the appendix.

5 SPDX Format

The content below describes the syntax that was used to generate an SPDX v2.2 conforming SBOM document for the Healthcare POC. Please refer to the SPDX website for additional background on SPDX formatting and conventions.

5.1 Document Creation Information

5.1.1 SBOM Document Identity

This section of the SPDX document contains the **SBOM Document Identity** and additional required content.

- *SPDXVersion* – fixed value of "SPDX-2.2"
- *DataLicense* – fixed value of "CC0-1.0"
- *SPDXID Document Identifier* – fixed value of "SPDXRef-Document"
- *DocumentName* – supplier document identifier, **SBOM Document Name**
- *DocumentNamespace* – supplier namespace to scope the document. This namespace plus the SPDXID make a unique SPDX identifier for the **Primary Component** and the **Included Components**.
- *Creator* – **Author Name**
- *Created* – created or effective date of the document, serves as the **SBOM Document Version**
- *CreatorComment* – supplier comments about the document

5.1.2 Notes

SPDX does not have field for the SBOM Document Version, so the Created tag was used:

- Created: 2020-10-07T11:26:24Z

The Created Date and Time can be added to the end of the DocumentName so that each SBOM document version will have a unique SBOM Document Name.

SPDX Document Creation Information Example

```
##Document Header
SPDXVersion: SPDX-2.2
DataLicense: CC0-1.0
SPDXID: SPDXRef-DOCUMENT
DocumentName:  ACME-INFUSION-1.0-SBOM-DRAFTv2020-10-08T180150
DocumentNamespace:  http://www.hospitalproducts.acme
Creator: Organization:  ACME-Hospital-Division()
Created: 2020-10-08T18:01:50Z
CreatorComment:  <text>Draft SCME INFUSION PoC II SBOM document in SPDX format. Unofficial content for demonstration purposes only.</text>
```

5.2 Package Information

The content for a **Primary Component** or an **Included Component** was captured as Package Information in an SPDX document.

5.2.1 Software Identity

The following describes the SPDX elements that capture the **Software Identity** information, as well as the SPDX content that identifies if the component is a **Primary Component** or an **Included Component**.

- *PackageName* – **Component Name**
- *SPDXID*
    - Construct an *SPDXID* using **Component Name** (PackageName) and **Version String**.
    - The *SPDXID* together with the *DocumentNamespace* satisfies the requirement for a **Unique Identifier** as defined by Framing. However, the identifier is primarily useful in the context of the SPDX documents (or for referring between SPDX documents). Therefore, an additional Unique Identifier using *purl* is also provided as described below.
- *PackageVersion* – **Version String**
- *PackageSupplier* – **Supplier Name**
- **Relationship**
    - Use the SPDX *Relationship* tag
    - Use the *SPDXID* tag to establish the relationship between two components
    - **Primary Component**
        - Defined using “DESCRIBES”
        - The **SPDX Document** describes the **Primary Component**
        - *Relationship*: *SPDXRef-DOCUMENT DESCRIBES <SPDXRef-1>*
    - **Included Component**
        - Defined using “CONTAINS”
        - The **Primary Component** contains the **Included Component**
        - *Relationship: <SPDXRef-1> CONTAINS <SPDXRef-2>*

5.2.2 Unique Identifier

The following SPDX conventions were used to describe the supplier-provided Unique Identifier established in Section 4.2 Unique Identifier in addition to the SPDX-required identifier.

- purl was used as a **Unique Identifier** that represents the supplier identifier
- a **purl** external reference identifier in SPDX was constructed using **Supplier Name**, **Component Name**, and **Version String**.
    - *ExternalRef: PACKAGE-MANAGER purl pkg:supplier/<**Supplier Name**>/**Component Name**>@<**Version String**>*

5.2.3 SBOM Completeness

The SPDX Relationship tag was used to provide the SBOM completeness of the component, according to the guidance of Section 4.3 SBOM Completeness. The following describes the SPDX elements and conventions used to document SBOM completeness in the Healthcare POC SBOMs. In the examples, the <Package SPDXID> is the SPDXID of the component that the statement covers.

- Unknown
    - *Relationship: SPDX-REF<Package SPDXID> CONTAINS NOASSERTION*
    - No additional "CONTAINS" relationships should be included in the SBOM
- Partial
    - *Relationship: SPDX-REF< Package SPDXID > CONTAINS NOASSERTION*
    - Additional "CONTAINS" relationships should be included
    - Or, an SBOM reference is included and the **Primary Component** is "Partial" in that SBOM
- Known
    - *Relationship: SPDX-REF< Package SPDXID > CONTAINS NONE*
    - One or more additional "CONTAINS" relationships should be included
    - Or, an SBOM reference is included and the **Primary Component** is "Known" in that SBOM
- Root
    - *Relationship: SPDX-REF< Package SPDXID > CONTAINS NONE*
    - No other "CONTAINS" relationships are included
    - Or, an SBOM Reference is included and the **Primary Component** is "Root" in that SBOM

5.2.4 Additional SPDX Elements

The following elements were required to generate a conforming SPDX document and not currently applicable for the Healthcare POC. Default values are used for the tags.

- PackageDownloadLocation: NOASSERTION
- FilesAnalyzed: false
- PackageLicenseConcluded: NOASSERTION
- PackageLicenseDeclared: NOASSERTION
- PackageCopyrightText: NOASSERTION

SPDX Package Information Example

```
##
## Packages
##
PackageName: INFUSION
SPDXID: SPDXRef-INFUSION-1.0
ExternalRef: PACKAGE-MANAGER purl pkg:supplier/ACME/INFUSION@1.0
PackageVersion: 1.0
PackageSupplier: Organization: ACME
Relationship: SPDXRef-DOCUMENT DESCRIBES SPDXRef-INFUSION-1.0
Relationship: SPDXRef-INFUSION-1.0 CONTAINS NOASSERTION
PackageDownloadLocation: NOASSERTION
FilesAnalyzed: false
PackageLicenseConcluded: NOASSERTION
PackageLicenseDeclared: NOASSERTION
PackageCopyrightText: NOASSERTION

PackageName: Bobs Browser
SPDXID: SPDXRef-Bobs-Browser-1
ExternalRef: PACKAGE-MANAGER purl pkg:supplier/Bob/Bobs-Browser@1
PackageVersion: 1
PackageSupplier: Organization: Bob
Relationship: SPDXRef-INFUSION-1.0 CONTAINS SPDXRef-Bobs-Browser-1
Relationship: SPDXRef-Bobs-Browser-1 CONTAINS NONE
PackageDownloadLocation: NOASSERTION
FilesAnalyzed: false
PackageLicenseConcluded: NOASSERTION
PackageLicenseDeclared: NOASSERTION
PackageCopyrightText: NOASSERTION
```

5.3 SBOM Content for Included Components

The following guidance was followed for capturing SBOM content for Included Components.

- Use the *Relationship* tag
- When including content in the SBOM document
  - The same syntax is used for first-level and next-level includes
  - *Relationship: SPDXRef<Package SPDXID> CONTAINS SPDXRef<Package SPDXID>*
- When referencing an external document
  - The external document is defined using the *ExternalDocumentRef* tag

   - *ExternalDocumentRef: DocumentRef-`[idstring] [SPDX Document URI] [Checksum]`*

- Use the *Relationship* tag to identify the external document containing the Package Information for the **Included Component**

   - *Relationship: SPDXRef-[idstring] CONTAINS ["DocumentRef-"[idstring]":"]SPDXID*

   - where "DocumentRef-"[idstring]":" is the external SPDX document

The system-of-system example in Appendix A contains examples of referencing an external document for the SBOM of an **Included Component** Examples.

5.4 SPDX Example

Below is a sample SBOM file in SPDX format for an ACME Infusion v1.0 pump that includes the component Bobs Browser v1.

SPDX Example for ACME Infusion pump

```
##Document Header
SPDXVersion: SPDX-2.2
DataLicense: CC0-1.0
SPDXID: SPDXRef-DOCUMENT
DocumentName:  ACME-INFUSION-1.0-SBOM-DRAFTv2020-10-08T180150
DocumentNamespace:  http://www.hospitalproducts.acme
Creator: Organization:  ACME-Hospital-Division()
Created: 2020-10-08T18:01:50Z
CreatorComment:  <text>Draft SCME INFUSION PoC II SBOM document in SPDX format. Unofficial content for demonstration purposes only.</text>


##
## Packages
##
PackageName: INFUSION
SPDXID: SPDXRef-INFUSION-1.0
ExternalRef: PACKAGE-MANAGER purl pkg:supplier/ACME/INFUSION@1.0
PackageVersion: 1.0
PackageSupplier: Organization: ACME
Relationship: SPDXRef-DOCUMENT DESCRIBES SPDXRef-INFUSION-1.0
Relationship: SPDXRef-INFUSION-1.0 CONTAINS NOASSERTION
PackageDownloadLocation: NOASSERTION
FilesAnalyzed: false
PackageLicenseConcluded: NOASSERTION
PackageLicenseDeclared: NOASSERTION
PackageCopyrightText: NOASSERTION

PackageName: Bobs Browser
SPDXID: SPDXRef-Bobs-Browser-1
ExternalRef: PACKAGE-MANAGER purl pkg:supplier/Bob/Bobs-Browser@1
PackageVersion: 1
PackageSupplier: Organization: Bob
Relationship: SPDXRef-INFUSION-1.0 CONTAINS SPDXRef-Bobs-Browser-1
Relationship: SPDXRef-Bobs-Browser-1 CONTAINS NONE
PackageDownloadLocation: NOASSERTION
FilesAnalyzed: false
PackageLicenseConcluded: NOASSERTION
PackageLicenseDeclared: NOASSERTION
PackageCopyrightText: NOASSERTION
```

6 SWID Format

SWID is an XML document composed of a structured set of data elements that identify the software product, product version, and organizational information. The SBOM baseline elements are mapped into the following SWID tags.

6.1 Baseline Components

The following SWID tags and conventions were used to establish the component information in the SBOM document.

- *xml:lang* – for language support. SBOM can be in local language
- *name*: Application/software name (**Component Name)**
- *tagID* – Globally unique identifier for the software **(Unique Identifier)**
- *version* – version of the software application/solution **(Version String)**
- *versionScheme* – Defines version string format
- *Entity name – Software creator name (***Supplier Name)**
- *regid – unique Id for the software creator* ***(Supplier ID)***
- *role – role of the entity e.g., tagCreator, softwareCreator* ***(Supplier role)***

The following SWID tags and conventions were used to establish the package dependency information.

- **Relationship**
  - link rel = relationship information to the primary component. Possible predefined values to attribute are **"component, feature, requires, etc."**
  - href = SWID of the referred/dependent component
  - use = dependency relationship ('required', 'recommended' -installed by default, 'optional' -excluded by default) (**Component Relationships**)

6.2 Unique Identifier

SWID tagId is a value that shall be globally unique for every SWID tag created. The SWID specification does not have any restrictions on tagId apart from globally unique identifier for each software component. One of the recommended unique value generators for tagId is the 16-byte GUID. Another method is to construct unique value is to use organization name, product name, version, edition, revision, etc. For the Healthcare POC, the **purl** string was used as unique value for tagId.

- The same **purl** convention as defined in SPDX Section 5.2.2 Unique Identifier was used for tagId
- tagId="pkg:supplier/<SupplierName>/<ComponentName>@<VersionString>"

6.3 SBOM Document Identity

SWID tagVersion attribute is used to identify the version of the SBOM document. Document version is used to update the version of the SBOM document to make a correction/clarification for the same version of the underlying software.

- **tagVersion** attribute was used to specify the SBOM document version.

6.4 SBOM Completeness

In SWID, the SBOM completeness information was documented in the "role" attribute of the "Entity" element with "authoritative" and "non-authoritative" tag information.

- In authoritative tags, the Entity's role attribute can have value of "tagCreator" and any one of "aggregator," "distributor," "licensor," or "softwareCreator" attributes
- In non-authoritative tags, the role attribute can have "tagCreator" value only

The authoritative and non-authoritative SWID tag information was used to satisfy the following SBOM completeness requirements.

- Unknown
  - SBOM was not collected, and SCA was not performed
  - e.g., <Entity name="The ACME Corporation" regid="acme.com" role="**tagCreator**"/>
  - The component has non-authoritative tag because the SBOM completeness is unknown
- Partial
  - Best-effort was used to identify included components or SCA was performed to identify included components
  - e.g., <Entity name="The ACME Corporation" regid="acme.com" role="**tagCreator distributor**"/>
  - The component has an authoritative tag that does not contain the "softwareCreator" value
- Known
  - All Included Components were identified
  - e.g., <Entity name="The ACME Corporation" regid="acme.com" role="**tagCreator softwareCreator** "/>
  - The component has an authoritative tag that contains the "softwareCreator" value. One or more link elements are contained in the <SoftwareIdentity> element
- Root
  - The Primary Component contains no Included Components
  - e.g., <Entity name="The ACME Corporation" regid="acme.com" role="**tagCreator softwareCreator** "/>
  - The component has an authoritative tag that contains the "softwareCreator" value. No additional link elements are contained in the <SoftwareIdentity> element

6.5 SBOM Content for Included Components

In SWID format, SBOM content for **Included Components** was provided with the XML document specification. All included components are defined within the <SoftwareIdentity></ SoftwareIdentity> with sub-element <Link></Link>

- Use <Link> element to show the relationship between the source tags and other applications

6.6 SWID Example

Below is sample SBOM file in SWID format for ACME Infusion pump V2.0 with following two components

- Windows 10 V1909
- SQL Server 2016

SWID file for ACME Infusion pump

```
<SoftwareIdentity
                name="Infusion pump"
                tagId="pkg:supplier/ACME/Infusionpump@2.0"
                version = "2.0"
                tagVersion="1">

        <Entity name="ACME Corporation"
                        regid = "acme.com" role="tagCreator softwareCreator"/>
        <!--Authoritative tag. SBOM of this component is KNOWN -->
        <!--For depedentent complete, check their respective SWID file -->

        <Link rel="component" href="swid: pkg:supplier/Microsoft/Windows10@1909" use ="required"/>
        <Link rel="component" href="swid: pkg:supplier/Microsoft/SQLServer2016@1.0" use ="required"/>
</SoftwareIdentity>
```

SWID file for Windows 10 V1909

```
<SoftwareIdentity
                name="Windows 10"
                tagId="pkg:supplier/Microsoft/Windows10@1909"
                version = "1909"
                tagVersion="1">

        <Entity name=”Microsoft Corporation”
                        regid = “microsoft.com” role=”tagCreator”/>
        <!—Non-authoritative tag. SBOM of this component is UNKNOWN -->
</SoftwareIdentity>
```

SWID file for SQL Server 2016

```
<SoftwareIdentity
                name="SQL Server 2016"
                tagId="pkg:supplier/Microsoft/SQLServer2016@1.0"
                version = "1.0"
                tagVersion="1">

        <Entity name=”Microsoft Corporation”
                        regid = “microsoft.com” role=”tagCreator”/>
        <!—Non-authoritative tag. SBOM of this component is UNKNOWN -->
</SoftwareIdentity>
```

7 Tools

Tools discovered, evaluated, and/or used during the Healthcare POC are described in this section.

7.1 SBOM Generation Tools

The SBOMs generated for the Healthcare POC were created manually, with tools developed as part of NTIA effort, and with commercially available products. The NTIA tools that were used are described in the following sub-sections.

7.1.1 NTIA SwiftBOM Web Tool

The Framing group developed the SwiftBOM Web tool to automate the generation of SBOMs. The tool supports the generation of multiple outputs and provides a visual representation of the SBOM.

The tool can be accessed with the following link:

SwiftBOM: https://sbom.democert.org/sbom/

7.1.2 NTIA Excel Tool

A SPDX generator in Excel was created that:

- Uses Excel formulas
- Is populated by uses with basic Document, Supplier, and Component information
- Automatically generates SPDX

The tool provides simple automation of the SBOM generation. A copy of the spreadsheet template is available through NTIA.

7.2 SBOM Conformance Tools

This section describes tools that were used to confirm the generated SBOM document conformed to the specification.

7.2.1 Website

A Web-based validator tool was used to validate the SPDX SBOMs. The tool discovers issues such as duplicate SPDXIDs and incomplete tag definitions. The error messages provide basic information that can be used to identify and correct issues with the format. The tool is located at:

https://tools.spdx.org/app/

7.2.2 Desktop Tool

SPDX tools and "how to use" documents are available at:

https://spdx.dev/resources/tools/

The following is useful information on using the SPDX desktop tool.

- Tools are available for most recent version of the SPDX specifications. For the SBOM proof of concept, we picked the latest version of SPDX, V2.2.2.
- Download the file spdx-tools-2.2.2-jar-with-dependencies.jar

- You need to run JRE 1.8 or later to run these tools. Please download and install JRE 1.8 or a later version.
- In the command line, run java -jar spdx-tools-2.2.2-jar-with-dependencies.jar. This will list all available options of the tool.
- To validate the created SPDX file, run the following in the command line:
  - ➢ java -jar spdx-tools-2.2.2-jar-with-dependencies.jar Verify “generated spdx filename”

  The tool will display the error message with the line number where the syntax is violating the specification. If there is no error, the tool confirms that the file is a valid SPDX document.
- A valid SPDX document can be converted into different formats like rdf, xls or html using this tool. Usage is self-explanatory.

7.3 Tooling Ecosystems

Tooling available for the different ecosystems has been captured based on the following categories:

Category	**Type**	**Description**
Produce	Build	The document is automatically created as part of building an artifact and contains information about the build.
	Manual	A person will manually fill in the information.
	Audit Tool	A source code analysis or audit tool will generate the document by inspection of the artifact and any associated sources.
Consume	View	Be able to understand the contents in human readable form (picture, figures, tables, text.). Use to support decision making & business processes.
	Diff	Be able to compare two documents of a given formation and clearly see the differences. For instance, comparing between two versions of a piece of software.
	Analyze	Be able to import a document into software. (e.g. Software Risk assessment)
Transform	Translate	Change from one file type to another file type while preserving the same information.
	Merge	Multiple sources of documents can be merged together for analysis and audit purposes
	Tool integration	Support use in other tools by APIs, libraries.

The sections below provide links to documents that capture the tooling for an ecosystem.

7.3.1 SPDX

- SPDX: https://tiny.cc/SPDX
- Google Doc: https://docs.google.com/document/d/1A1jFIYihB-IyT0gv7E_KoSjLbwNGmu_wOXBs6siemXA/edit

7.3.2 SWID

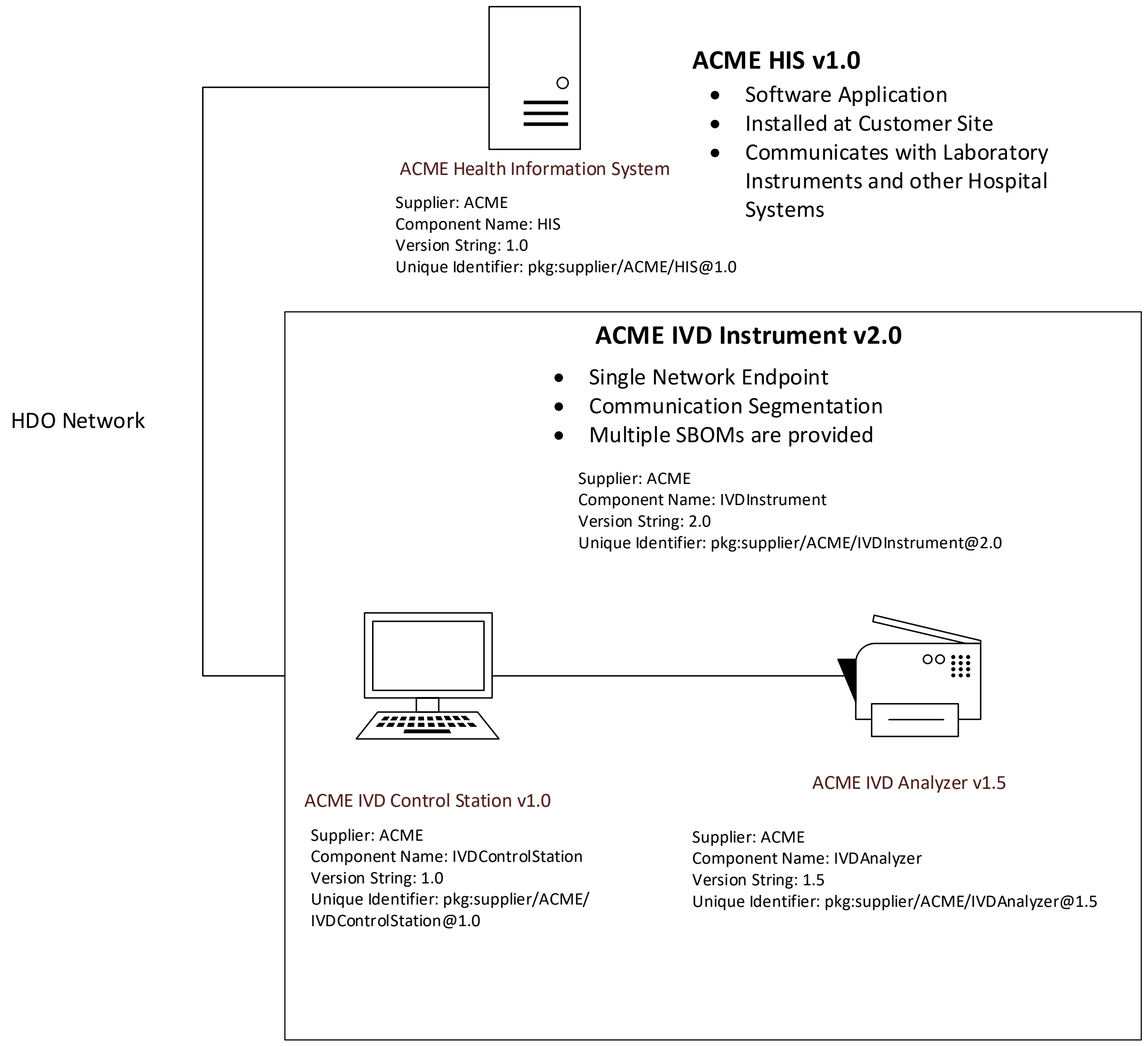
- SWID: http://tiny.cc/SWID
- Google Doc: https://docs.google.com/document/d/1oebYvHcOhtMG8Uhnd5he0l_vhty7MsTjp6fYCOwUmwM/edit

Appendix A: System of Systems Example

The following is a system-of-system example for a single endpoint represented by the ACME IVD Instrument. The diagram and SBOM content cover the following:

- Health Information System v1.0
    - A product that is distributed separately and communicates with the ACMD IVD Instrument, as well as other instrument platforms
    - ACME provides an SBOM for the ACME HIS v1.0 product
- ACME IVD Instrument v2.0
    - A medical instrument that contains two hardware subsystems
    - ACME provides an SBOM for the medical device and each subsystem
        - ACME IVD Instrument V2.0
            - Includes an external reference to ACME IVD Control Station
            - Includes an external reference to ACME IVD Analyzer
        - ACME IVD Control Station
        - ACME IVD Analyzer

System of System with Single Endpoint Diagram


ACME HIS v1.0
Software Application
Installed at Customer Site
Communicates with Laboratory Instruments and other Hospital Systems
ACME Health Information System
Supplier: ACME
Component Name: HIS
Version String: 1.0
Unique Identifier: pkg:supplier/ACME/HIS@1.0
HDO Network
ACME IVD Instrument v2.0
Single Network Endpoint
Communication Segmentation
Multiple SBOMs are provided
Supplier: ACME
Component Name: IVDInstrument
Version String: 2.0
Unique Identifier: pkg:supplier/ACME/IVDInstrument@2.0
ACME IVD Control Station v1.0
Supplier: ACME
Component Name: IVDControlStation
Version String: 1.0
Unique Identifier: pkg:supplier/ACME/IVDControlStation@1.0
ACME IVD Analyzer v1.5
Supplier: ACME
Component Name: IVDAnalyzer
Version String: 1.5
Unique Identifier: pkg:supplier/ACME/IVDAnalyzer@1.5

1. SPDX SBOMs

SPDX content for HIS

```
##Document Header
SPDXVersion: SPDX-2.2
DataLicense: CC0-1.0
SPDXID: SPDXRef-DOCUMENT
DocumentName:  ACME-HIS-1.0-SBOMv2020-10-08T180804
DocumentNamespace:  http://www.hospitalproducts.acme
Creator: Organization:  ACME-Hospital-Division()
Created: 2020-10-08T18:08:04Z
CreatorComment:  <text>Draft ACME HIS 1.0 PoC II SBOM document in SPDX format. Unofficial content for demonstration purposes only.</text>


##
## Packages
##
PackageName: HIS
SPDXID: SPDXRef-HIS-1.0
ExternalRef: PACKAGE-MANAGER purl pkg:supplier/ACME/HIS@1.0
PackageVersion: 1.0
PackageSupplier: Organization: ACME
Relationship: SPDXRef-DOCUMENT DESCRIBES SPDXRef-HIS-1.0
Relationship: SPDXRef-HIS-1.0 CONTAINS NOASSERTION
PackageDownloadLocation: NOASSERTION
FilesAnalyzed: false
PackageLicenseConcluded: NOASSERTION
PackageLicenseDeclared: NOASSERTION
PackageCopyrightText: NOASSERTION
```

SPDX content for IVD Instrument V2.0

```
##Document Header
SPDXVersion: SPDX-2.2
DataLicense: CC0-1.0
SPDXID: SPDXRef-DOCUMENT
DocumentName:  ACME-IVDInstrument-2.0-SBOMv2020-10-08T181346
DocumentNamespace:  http://www.hospitalproducts.acme
Creator: Organization:  ACME-Hospital-Division()
Created: 2020-10-08T18:13:46Z
CreatorComment:  <text>Draft ACME IVD Instrument 2.0 PoC II SBOM document in SPDX format. Unofficial content for
demonstration purposes only.</text>


##
## Packages
##
PackageName: IVDInstrument
SPDXID: SPDXRef-IVDInstrument-2.0
ExternalRef: PACKAGE-MANAGER purl pkg:supplier/ACME/IVDInstrument@2.0
PackageVersion: 2.0
PackageSupplier: Organization: ACME
Relationship: SPDXRef-DOCUMENT DESCRIBES SPDXRef-IVDInstrument-2.0
Relationship: SPDXRef-IVDInstrument-2.0 CONTAINS NONE
PackageDownloadLocation: NOASSERTION
FilesAnalyzed: false
PackageLicenseConcluded: NOASSERTION
PackageLicenseDeclared: NOASSERTION
PackageCopyrightText: NOASSERTION

PackageName: IVDControlStation
SPDXID: SPDXRef-IVDControlStation-1.0
ExternalRef: PACKAGE-MANAGER purl pkg:supplier/ACME/IVDControlStation@1.0
PackageVersion: 1.0
PackageSupplier: Organization: ACME
Relationship: SPDXRef-IVDInstrument-2.0 CONTAINS SPDXRef-IVDControlStation-1.0
Relationship: SPDXRef-IVDControlStation-1.0 CONTAINS NOASSERTION
PackageDownloadLocation: NOASSERTION
FilesAnalyzed: false
PackageLicenseConcluded: NOASSERTION
PackageLicenseDeclared: NOASSERTION
PackageCopyrightText: NOASSERTION
ExternalDocumentRef: DocumentRef-ACME-IVDControlStation-1.0-SBOMv2020-10-07T205156
http://www.hospitalproducts.acme SHA1: adaca62f7013ea724b1ca20083dfa63a1ade53b8
Relationship:  SPDXRef-IVDControlStation-1.0 CONTAINS DocumentRef-ACME-IVDControlStation-1.0-SBOMv2020-10-
07T205156:SPDXRef-IVDControlStation-1.0
```

SPDX content for IVD Instrument V2.0 (cont.)

```
PackageName: IVDAnalyzer
SPDXID: SPDXRef-IVDAnalyzer-1.5
ExternalRef: PACKAGE-MANAGER purl pkg:supplier/ACME/IVDAnalyzer@1.5
PackageVersion: 1.5
PackageSupplier: Organization: ACME
Relationship: SPDXRef-IVDInstrument-2.0 CONTAINS SPDXRef-IVDAnalyzer-1.5
Relationship: SPDXRef-IVDAnalyzer-1.5 CONTAINS NONE
PackageDownloadLocation: NOASSERTION
FilesAnalyzed: false
PackageLicenseConcluded: NOASSERTION
PackageLicenseDeclared: NOASSERTION
PackageCopyrightText: NOASSERTION
ExternalDocumentRef: DocumentRef-ACME-IVD-Analyzer-1.5-SBOMv2020-10-07T204107 http://www.hospitalproducts.acme
SHA1: 9d40bcac4b1d97c96a24e7208106a14bab57b0da
Relationship:  SPDXRef-IVDAnalyzer-1.5 CONTAINS DocumentRef-ACME-IVD-Analyzer-1.5-SBOMv2020-10-
07T204107:SPDXRef-IVDAnalyzer-1.5
```

SPDX content for IVD Control Station V1.0

```
##Document Header
SPDXVersion: SPDX-2.2
DataLicense: CC0-1.0
SPDXID: SPDXRef-DOCUMENT
DocumentName:  ACME-IVDControlStation-1.0-SBOMv2020-10-08T181847
DocumentNamespace:  http://www.hospitalproducts.acme
Creator: Organization:  ACME-Hospital-Division()
Created: 2020-10-08T18:18:47Z
CreatorComment:  <text>Draft ACME IVD Control Station PoC II SBOM document in SPDX format. Unofficial content for
demonstration purposes only.</text>


##
## Packages
##
PackageName: IVDControlStation
SPDXID: SPDXRef-IVDControlStation-1.0
ExternalRef: PACKAGE-MANAGER purl pkg:supplier/ACME/IVDControlStation@1.0
PackageVersion: 1.0
PackageSupplier: Organization: ACME
Relationship: SPDXRef-DOCUMENT DESCRIBES SPDXRef-IVDControlStation-1.0
Relationship: SPDXRef-IVDControlStation-1.0 CONTAINS NOASSERTION
PackageDownloadLocation: NOASSERTION
FilesAnalyzed: false
PackageLicenseConcluded: NOASSERTION
PackageLicenseDeclared: NOASSERTION
PackageCopyrightText: NOASSERTION

PackageName: Windows_7
SPDXID: SPDXRef-Windows_7-SP1
ExternalRef: PACKAGE-MANAGER purl pkg:supplier/Microsoft/Windows_7@SP1
PackageVersion: SP1
PackageSupplier: Organization: Microsoft
Relationship: SPDXRef-IVDControlStation-1.0 CONTAINS SPDXRef-Windows_7-SP1
Relationship: SPDXRef-Windows_7-SP1 CONTAINS NONE
PackageDownloadLocation: NOASSERTION
FilesAnalyzed: false
PackageLicenseConcluded: NOASSERTION
PackageLicenseDeclared: NOASSERTION
PackageCopyrightText: NOASSERTION
```

SPDX content for IVD Control Station V1.0 (cont.)

```
PackageName: SQL 2005 Express
SPDXID: SPDXRef-SQL-2005-Express-9.00.5000.00,SP4
ExternalRef: PACKAGE-MANAGER purl pkg:supplier/Microsoft/SQL-2005-Express@9.00.5000.00,SP4
PackageVersion: 9.00.5000.00,SP4
PackageSupplier: Organization: Microsoft
Relationship: SPDXRef-IVDControlStation-1.0 CONTAINS SPDXRef-SQL-2005-Express-9.00.5000.00,SP4
Relationship: SPDXRef-SQL-2005-Express-9.00.5000.00,SP4 CONTAINS NOASSERTION
PackageDownloadLocation: NOASSERTION
FilesAnalyzed: false
PackageLicenseConcluded: NOASSERTION
PackageLicenseDeclared: NOASSERTION
PackageCopyrightText: NOASSERTION

PackageName: .Net Frame Work
SPDXID: SPDXRef-.Net-Frame-Work-V2.1.21022.8,SP2
ExternalRef: PACKAGE-MANAGER purl pkg:supplier/Microsoft/.Net-Frame-Work@V2.1.21022.8,SP2
PackageVersion: V2.1.21022.8,SP2
PackageSupplier: Organization: Microsoft
Relationship: SPDXRef-IVDControlStation-1.0 CONTAINS SPDXRef-.Net-Frame-Work-V2.1.21022.8,SP2
Relationship: SPDXRef-.Net-Frame-Work-V2.1.21022.8,SP2 CONTAINS NOASSERTION
PackageDownloadLocation: NOASSERTION
FilesAnalyzed: false
PackageLicenseConcluded: NOASSERTION
PackageLicenseDeclared: NOASSERTION
PackageCopyrightText: NOASSERTION

PackageName: Bobs Browser
SPDXID: SPDXRef-Bobs-Browser-1
ExternalRef: PACKAGE-MANAGER purl pkg:supplier/ArtMan/Bobs-Browser@1
PackageVersion: 1
PackageSupplier: Organization: ArtMan
Relationship: SPDXRef-IVDControlStation-1.0 CONTAINS SPDXRef-Bobs-Browser-1
Relationship: SPDXRef-Bobs-Browser-1 CONTAINS NONE
PackageDownloadLocation: NOASSERTION
FilesAnalyzed: false
PackageLicenseConcluded: NOASSERTION
PackageLicenseDeclared: NOASSERTION
PackageCopyrightText: NOASSERTION
```

SPDX Content for IVD Analyzer V1.5

```
##Document Header
SPDXVersion: SPDX-2.2
DataLicense: CC0-1.0
SPDXID: SPDXRef-DOCUMENT
DocumentName:  ACME-IVD-Analyzer-1.5-SBOMv2020-10-08T182447
DocumentNamespace:  http://www.hospitalproducts.acme
Creator: Organization:  ACME-Hospital-Division()
Created: 2020-10-08T18:24:47Z
CreatorComment:  <text>Draft ACME IVD Analyzer PoC II SBOM document in SPDX format. Unofficial content for
demonstration purposes only.</text>


##
## Packages
##
PackageName: IVDAnalyzer
SPDXID: SPDXRef-IVDAnalyzer-1.5
ExternalRef: PACKAGE-MANAGER purl pkg:supplier/ACME/IVDAnalyzer@1.5
PackageVersion: 1.5
PackageSupplier: Organization: ACME
Relationship: SPDXRef-DOCUMENT DESCRIBES SPDXRef-IVDAnalyzer-1.5
Relationship: SPDXRef-IVDAnalyzer-1.5 CONTAINS NONE
PackageDownloadLocation: NOASSERTION
FilesAnalyzed: false
PackageLicenseConcluded: NOASSERTION
PackageLicenseDeclared: NOASSERTION
PackageCopyrightText: NOASSERTION
```

2. SWID SBOMs

SWID content for HIS

```
<SoftwareIdentity
                name="HIS"
                tagId="pkg:supplier/ACME/HIS@1.0"
                version = "1.0"
                tagVersion ="1">

        <Entity name="ACME Corporation"
                        regid = "acme.com" role="tagCreator"/>
        <!-- Non-authoritative tag. SBOM status us UNKNOWN -->
</SoftwareIdentity>
```

SWID content for IVD Instrument V2.0

```
<SoftwareIdentity
                name="IVD Instrument"
                tagId="pkg:supplier/ACME/IVDInstrument@2.0"
                version = "2.0"
                tagVersion="1">

        <Entity name="ACME Corporation"
                        regid = "acme.com" role="tagCreator softwareCreator"/>
        <!--Authoritative tag. SBOM of this component is KNOWN. SBOM for this component is good -->
        <!--For depedentent complete, check their respective SWID file -->

        <Link rel="component" href="swid: pkg:supplier/ACME/IVDControlStation@1.0" use ="required"/>
        <Link rel="component" href="swid: pkg:supplier/ACME/IVDAnalyzer@1.5" use ="required"/>
</SoftwareIdentity>
```

SWID content for IVD Control Station V1.0

```
<SoftwareIdentity
                name="IVD Control Station"
                tagId="pkg:supplier/ACME/IVDControlStation@1.0"
                version = "1.0"
                tagVersion="1">

        <Entity name=”ACME Corporation”
                        regid = “acme.com” role=”tagCreator softwareCreator”/>
        <!--Authoritative tag. SBOM of this component is KNOWN -->

        <Link rel="component" href="swid: pkg:supplier/Microsoft/Windows_7@SP1 use="required"/>
        <Link rel="component" href="swid: pkg:supplier/Microsoft/SQL-2005-Express@9.00.5000.00-SP4 use ="required"/>
        <Link rel="component" href="swid: pkg:supplier/Microsoft/.Net-Frame-Work@V2.1.21022.8-SP2 use ="required"/>
        <Link rel="component" href="swid: pkg:supplier/ArtMan/Bobs-Browser@1-SP2 use ="required"/>
</SoftwareIdentity>
```

SWID contnet for IVD Analyzer V1.5

```
<SoftwareIdentity
                name="IVD Analyzer"
                tagId="pkg:supplir/ACME/IVDAnalyzer@1.5"
                version = "1.5"
                tagVersion="1">

        <Entity name="ACME Corporation"
                        regid = "acme.com" role="tagCreator softwareCreator"/>
        <!--Authoritative tag. SBOM of this component is ROOT -->
</SoftwareIdentity>
```

SWID content for Windows 7 SP1

```
<SoftwareIdentity
                name="Windows 7 SP1"
                tagId="pkg:supplier/Microsoft/Windows_7@SP1"
                version = "SP1"
                tagVersion="1">

        <Entity name="Microsoft Corporation"
                        regid = "microsoft.com" role="tagCreator"/>
        <!--Non-authoritative tag. SBOM status us UNKNOWN -->
</SoftwareIdentity>
```

SWID content for SQL 2005 Express

```
<SoftwareIdentity
                name="SQL 2005 Express"
                tagId="pkg:supplier/Microsoft/SQL-2005-Express@9.00.5000.00-SP4"
                version = "9.00.5000.00-SP4"
                tagVersion="1">

        <Entity name="Microsoft Corporation"
                        regid = "microsoft.com" role="tagCreator"/>
        <!--Non-authoritative tag. SBOM status us UNKNOWN -->
</SoftwareIdentity>
```

SWID content for .NET Framework
```
<SoftwareIdentity
                    name=".NET Framework"
                    tagId="pkg:supplier/Microsoft/.Net-Frame-Work@V2.1.21022.8-SP2"
                    version = "2.1.21022.8-SP2"
                    tagVersion="1">

          <Entity name="Microsoft Corporation"
                              regid = "microsoft.com" role="tagCreator"/>
          <!--Non-authoritative tag. SBOM status us UNKNOWN -->
</SoftwareIdentity>
```

SWID content for Bobs Browser
```
<SoftwareIdentity
                    name="Bobs Browser"
                    tagId="pkg:supplier/ArtMan/Bobs-Browser@1.0"
                    version="1.0"
                    tagVersion="1">

          <Entity name="ArtMan"
                              regid = "microsoft.com" role="tagCreator"/>
          <!--Non-authoritative tag. SBOM status us UNKNOWN -->
</SoftwareIdentity>
```