ecma
INTERNATIONAL

Standard ECMA-424

1st Edition / June 2024

CycloneDX Bill of materials specification

Rue du Rhône 114   CH-1204 Geneva   T: +41 22 849 6000   F: +41 22 849 6001

Contents

Page

Introduction

CycloneDX is a modern standard designed to address the complexities of the software and system supply chain. Originating in 2017, CycloneDX has grown into a general-purpose Bill of Materials (BOM) standard capable of representing various types of inventories, including software, hardware, and services. CycloneDX continuously evolves to meet the changing needs of the industry, incorporating new features and improvements to stay ahead of emerging challenges.

The design philosophy of CycloneDX emphasizes simplicity and ease of use, making it accessible to both technical and non-technical stakeholders. Despite its straightforward design, CycloneDX is a full-stack BOM format with advanced capabilities. Its guiding principles include easy adoption, rapid risk identification, continuous improvement, and high degrees of automation and extensibility.

CycloneDX plays a crucial role in enhancing software and system transparency, providing detailed information about the components used in an application, including their versions, suppliers, and dependencies. This transparency is essential for identifying and managing risks, ensuring regulatory compliance, and building trust in both software and hardware systems. By offering a comprehensive and standardized way to document these components, CycloneDX enables organizations to achieve greater security and reliability in their supply chains, supporting a wide range of use cases from product security to vendor risk management.

This Ecma Standard was developed by Technical Committee 54 and was adopted by the General Assembly of June 2024.

CycloneDX v1.6 Bill of materials specification

1 Scope

This standard defines the CycloneDX v1.6 Bill of materials specification.

2 Conformance

2.1 Requirements Terminology

In this standard, the words that are used to define the significance of each requirement are detailed below. These words are used in accordance with their definitions in RFC 2119, and their respective meanings are reproduced below:

- Must: This word, or the adjective “required” and the auxiliary verb "shall", means that the item is an absolute requirement of the standard.
- Should: This word, or the adjective “recommended”, means that there might exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before making an implementation decision.
- May: This word, or the adjective “optional”, means that this item is truly optional.

The words "must not", "shall not", "should not", and "not recommended", are the negative forms of "must", "shall", "should", and "recommended", respectively. There is no negative form of "may".

2.2 Implementation Conformance

This standard includes the implementation requirements that systems processing CycloneDX content must satisfy in order to achieve conforming interoperability. An implementation is a consumer, or a producer, or both a consumer and a producer.

In order for a consumer to be considered conformant, the following rules apply:

- It must interpret and process the contents of CycloneDX BOMs in a manner conforming to this standard. A consumer is not required to interpret or process all of the content in a CycloneDX BOM.
- It should instantiate a warning or error condition when a CycloneDX BOM is not conforming to this standard.
- It must not instantiate an error condition in response to a CycloneDX BOM conforming to this standard.
- When optional or recommended features contained within CycloneDX BOMs are accessed by a consumer, the consumer must interpret and process those features in a manner conforming to this standard.

In order for a producer to be considered conformant, the following rules apply:

- Any CycloneDX BOM it creates must conform to this standard.
- It must not introduce any non-conforming CycloneDX content when modifying or enriching a CycloneDX BOM.
- When a producer chooses to use an optional or recommended feature in an CycloneDX BOM, then the producer must create or modify that feature in a manner conforming to this standard.

3 Normative References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ECMA ECMA-404, *The JSON Data Interchange Format* https://ecma-international.org/publications-and-standards/standards/ecma-404/

IETF RFC3339, *Date and Time on the Internet: Timestamps* https://tools.ietf.org/html/rfc3339

IETF RFC3986, *Uniform Resource Identifier (URI): Generic Syntax* https://tools.ietf.org/html/rfc3986

IETF RFC3987, *Internationalized Resource Identifiers (IRIs)* https://tools.ietf.org/html/rfc3987

W3C XML 1.1, *Extensible Markup Language (XML) 1.1 (Second Edition)* http://www.w3.org/TR/2006/REC-xml11-20060816/

W3C XML Schema 1.0, *XML Schema Part 1: Structures Second Edition* https://www.w3.org/TR/xmlschema-1/

4 Terms and Definitions

For the purposes of this standard, the following definitions apply. Terms explicitly defined in this standard are not to be presumed to refer implicitly to similar terms defined elsewhere.

4.1
Attestation
A formal declaration that something is true or accurate, often backed by documentation or verification from an authoritative source. It serves as a confirmation or proof of a fact, condition, or compliance with specific standards or requirements.

4.2
Author
A person who creates written works, such as software or data.

4.3
Component function
The purpose for which a software component exists. Examples of component functions include parsers, database persistence, and authentication providers.

4.4
Component type
The general classification of a software components architecture. Examples of component types include libraries, frameworks, applications, containers, and operating systems.

4.5
Manufacturer
An entity that develops and produces products such as virtual or physical goods.

4.6
Direct dependency
A component that is referenced by a main (metadata) component itself.

4.7
Package URL (PURL)
An ecosystem-agnostic specification which standardizes the syntax and location information of software components.

4.8
Pedigree
Data which describes the lineage and/or process for which software has been created or altered.

4.9
Procurement
The process of agreeing to terms and acquiring software or services for later use.

4.10
Provenance
The chain of custody and origin of a software component. Provenance incorporates the point of origin through distribution as well as derivatives in the case of software that has been modified.

4.11
Provider
An entity that offers services, infrastructure, or platforms. These services can include computing resources, storage, software applications, and networking capabilities.

4.12
Publisher
An entity that produces and distributes content, such as software, to the public.

4.13
Software Identification (SWID)
An ISO standard that formalizes how software is tagged.

4.14
Software Package Data Exchange (SPDX)
A Linux Foundation project which produces a standardized list of open source licences and defines an expression language for those licences.

4.15
Supplier
An entity that provides products or services to another entity, typically within a supply chain.

4.16
Third-party component
Any software component not directly created including open source, "source available", and commercial or proprietary software.

4.17
Transitive dependency
A software component that is indirectly used by another component by means of being a dependency of a dependency.

5 Overview of CycloneDX

CycloneDX is a modern standard for the software supply chain. At its core, CycloneDX is a general-purpose Bill of Materials (BOM) standard capable of representing software, hardware, services, and other types of inventory. The CycloneDX standard began in 2017 in the Open Worldwide Application Security Project (OWASP) community. CycloneDX is an OWASP flagship project, has a formal standardization process and governance model, and is supported by the global information security community.

5.1.1 Design Philosophy and Guiding Principles

The simplicity of design is at the forefront of the CycloneDX philosophy. The format is easily understandable by a wide range of technical and non-technical roles. CycloneDX is a full-stack BOM format with many advanced capabilities that are achieved without sacrificing the design philosophy. Some guiding principles influencing its design include:

- Be easy to adopt and easy to contribute to
- Identify risk to as many adopters as possible, as quickly as possible
- Avoid blockers that prevent the identification of risk
- Continuous improvement - innovate quickly and improve over time
- Encourage innovation and competition through extensions
- Produce immutable and backward-compatible releases

- Focus on high degrees of automation
- Provide a smooth path to specification compliance through prescriptive design

5.1.2 Defining Software Bill of Materials

The U.S. National Telecommunications and Information Administration (NTIA) defines software bill as materials as "*a formal, machine-readable inventory of software components and dependencies, information about those components, and their hierarchical relationships*." OWASP CycloneDX implements this definition and extends it in many ways, including adding services as a foundational component in a Software Bill of Materials.

5.1.3 The Role of SBOM in Software Transparency

Software transparency involves providing clear and accurate information about the components used in an application, including their name, version, supplier, and any dependencies required by the component. This information helps identify and manage the risks associated with the software whilst also enabling compliance with relevant regulations and standards. With the growing importance of software in our daily lives, transparency is critical to building trust in software and ensuring that it is safe, secure, and reliable.

SBOMs are the vehicle through which software transparency can be achieved. With SBOMs, parties throughout the software supply chain can leverage the information within to enable various use cases that would not otherwise be easily achievable. SBOMs play a vital role in promoting software transparency, allowing users to make informed decisions about the software they use.

5.2 High-Level SBOM Use Cases

A complete and accurate inventory of all first-party and third-party components is essential for risk identification. SBOMs should ideally contain all direct and transitive components and the dependency relationships between them.

CycloneDX far exceeds the Minimum Elements for Software Bill of Materials as defined by the National Telecommunications and Information Administration (NTIA) in response to U.S. Executive Order 14028.

Adopting CycloneDX allows organizations to quickly meet these minimum requirements and mature into using more sophisticated use cases over time. CycloneDX is capable of achieving all SBOM requirements defined in the OWASP Software Component Verification Standard (SCVS).

A few high-level use cases for SBOM include:

- Product security, architectural, and licence risk
- Procurement and M&A
- Software component transparency
- Supply chain transparency
- Vendor risk management

5.3 xBOM Capabilities

CycloneDX provides advanced supply chain capabilities for cyber risk reduction. Among these capabilities are:

- Software Bill of Materials (SBOM)
- Software-as-a-Service Bill of Materials (SaaSBOM)
- Hardware Bill of Materials (HBOM)
- Machine Learning Bill of Materials (ML-BOM)
- Cryptography Bill of Materials (CBOM)
- Operations Bill of Materials (OBOM)
- Manufacturing Bill of Materials (MBOM)
- Bill of Vulnerabilities (BOV)
- Vulnerability Disclosure Report (VDR)

- Vulnerability Exploitability eXchange (VEX)
- CycloneDX Attestations (CDXA)
- Common Release Notes Format

5.3.1 Software Bill of Materials (SBOM)

SBOMs describe the inventory of software components and services and the dependency relationships between them. A complete and accurate inventory of all first-party and third-party components is essential for risk identification. SBOMs should ideally contain all direct and transitive components and the dependency relationships between them.

5.3.2 Software-as-a-Service BOM (SaaSBOM)

SaaSBOMs provide an inventory of services, endpoints, and data flows and classifications that power cloud-native applications. CycloneDX is capable of describing any type of service, including microservices, Service Orientated Architecture (SOA), Function as a Service (FaaS), and System of Systems.

SaaSBOMs complement Infrastructure-as-Code (IaC) by providing a logical representation of a complex system, complete with an inventory of all services, their reliance on other services, endpoint URLs, data classifications, and the directional flow of data between services. Optionally, SaaSBOMs may also include the software components that make up each service.

5.3.3 Hardware Bill of Materials (HBOM)

CycloneDX supports many types of components, including hardware devices, making it ideal for use with consumer electronics, IoT, ICS, and other types of embedded devices. CycloneDX fills an important role in between traditional eBOM and mBOM use cases for hardware devices.

5.3.4 Machine Learning Bill of Materials (ML-BOM)

ML-BOMs provide transparency for machine learning models and datasets, which provide visibility into possible security, privacy, safety, and ethical considerations. CycloneDX standardizes model cards in a way where the inventory of models and datasets can be used independently or combined with the inventory of software and hardware components or services defined in HBOMs, SBOMs, and SaaSBOMs.

5.3.5 Cryptography Bill of Materials (CBOM)

A Cryptography Bill of Materials (CBOM) describes cryptographic assets and their dependencies. Discovering, managing, and reporting on cryptographic assets is necessary as the first step on the migration journey to quantum-safe systems and applications. Cryptography is typically buried deep within components used to compose and build systems and applications. As part of an agile cryptographic approach, organizations should seek to understand what cryptographic assets they are using and facilitate the assessment of the risk posture to provide a starting point for mitigation.

5.3.6 Operations Bill of Materials (OBOM)

OBOMs provide a full-stack inventory of runtime environments, configurations, and additional dependencies. CycloneDX is a full-stack bill of materials standard supporting entire runtime environments consisting of hardware, firmware, containers, operating systems, applications, and libraries. Coupled with the ability to specify configuration makes CycloneDX ideal for Operations Bill of Materials.

5.3.7 Manufacturing Bill of Materials (MBOM)

CycloneDX can describe declared and observed formulations for reproducibility throughout the product lifecycle of components and services. This advanced capability provides transparency into how components were made, how a model was trained, or how a service was created or deployed. In addition, every component and service in a CycloneDX BOM can optionally specify formulation and do so in existing BOMs or in dedicated MBOMs. By externalizing formulation into dedicated MBOMs, SBOMs can link to MBOMs for their components and services, and access control can be managed independently. This allows organizations to maintain tighter control over what parties gain access to inventory information in a BOM and what parties have access to MBOM information which may have higher sensitivity and data classification.

5.3.8 Bill of Vulnerabilities (BOV)

CycloneDX BOMs may consist solely of vulnerabilities and thus can be used to share vulnerability data between systems and sources of vulnerability intelligence. Complex vulnerability data can be represented, including the vulnerability source, references, multiple severities, risk ratings, details and recommendations, and the affected software and hardware, along with their versions.

5.3.9 Vulnerability Disclosure Report (VDR)

VDRs communicate known and unknown vulnerabilities affecting components and services. Known vulnerabilities inherited from the use of third-party and open-source software can be communicated with CycloneDX. Previously unknown vulnerabilities affecting both components and services may also be disclosed using CycloneDX, making it ideal for Vulnerability Disclosure Report (VDR) use cases. CycloneDX exceeds the data field requirements defined in ISO/IEC 29147:2018 for vulnerability disclosure information.

5.3.10 Vulnerability Exploitability eXchange (VEX)

VEX conveys the exploitability of vulnerable components in the context of the product in which they're used. VEX is a subset of VDR. Oftentimes, products are not affected by a vulnerability simply by including an otherwise vulnerable component. VEX allows software vendors and other parties to communicate the exploitability status of vulnerabilities, providing clarity on the vulnerabilities that pose a risk and the ones that do not.

5.3.11 CycloneDX Attestations (CDXA)

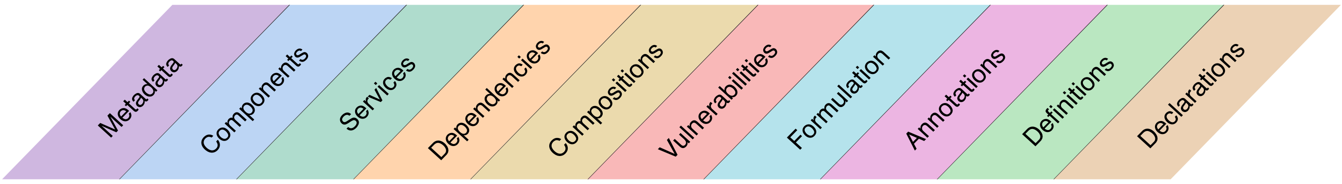
CycloneDX Attestations enable organizations to communicate security standards, claims, and evidence about security requirements, and attestations to the veracity and completeness of those claims. CycloneDX Attestations is a way to manage "compliance as code."

5.3.12 Common Release Notes Format

CycloneDX standardizes release notes into a common, machine-readable format. This capability unlocks new workflow potential for software publishers and consumers alike. This functionality works with or without the Bill of Materials capabilities of the specification.

5.4 CycloneDX Object Model

Within the root element, CycloneDX defines the following object types:



The object types are arranged in order and contain (but are not limited to) the following types of data:

Metadata	Supplier	Authors	Component		
	Manufacturer	Tools	Lifecycles		
Components	Supplier	Identity	Pedigree	Provenance	Evidence
	Component Type	Licenses	Hashes	Release Notes	Relationships
Services	Provider	Data Classification	Trust Zone		
	Endpoints	Data Flow	Relationships		
Dependencies	Components	Services			
Compositions	Completeness of: Components	Services	Dependencies	Vulnerabilities	
Vulnerabilities	Details	Source	Exploitability (VEX)	Targets Affected	Proof of Concept
	Advisories	Risk Ratings	Evidence	Version Ranges	Recommendations
Formulation	Declared	Formulas	Tasks	Components	
	Observed	Workflows	Steps	Services	
Annotations	Per Person	Per Organization	Per Tool		
	Details	Timestamp	Signature		
Definitions	Standards	Requirements	Levels		
Declarations	Attestations	Evidence	Conformance	Mitigation Strategies	Assessors
	Claims	Counter Evidence	Confidence	Signatories	Signatures
Extensions	Properties	Per Organization	Per Team		
	Formal Taxonomy	Per Industry	...		

5.4.1 BOM Identity

The bom element has properties for serialNumber and version. Together these two properties form the identity of a BOM. A BOM's identity can be expressed using a BOM-Link, a formally registered URN capable of referencing a BOM or any component, service, or vulnerability in a BOM. Refer to the chapter on Relationships for more information.

5.4.1.1 Serial Number

Every BOM generated should have a unique serial number, even if the contents of the BOM have not changed over time. If specified, the serial number must conform to RFC-4122. The use of serial numbers is recommended.

5.4.1.2 Version

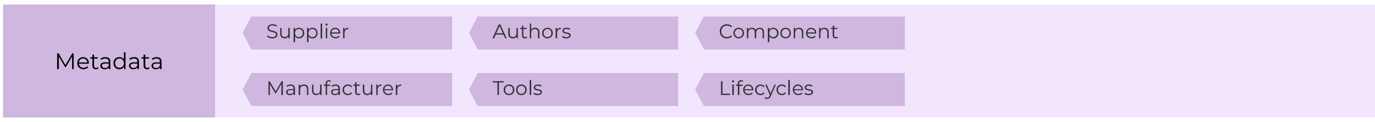
Whenever an existing BOM is modified, either manually or through automated processes, the version of the BOM should be incremented by 1. When a system is presented with multiple BOMs with identical serial numbers, the system should use the most recent version of the BOM. The default version is '1'.

5.4.2 The Anatomy of a CycloneDX BOM

The following are descriptions of the root-level elements of a CycloneDX BOM.

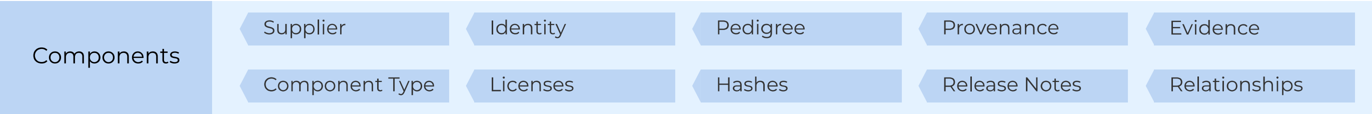
5.4.2.1 Metadata

BOM metadata includes the supplier, manufacturer, and target component for which the BOM describes. It also includes the tools used to create the BOM, and licence information for the BOM document itself.



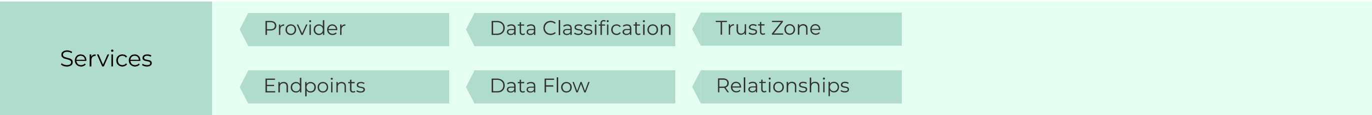
5.4.2.2 Components

Components describe the complete inventory of first-party and third-party components. The specification can represent software, hardware devices, machine learning models, source code, and configurations, along with the manufacturer information, licence and copyright details, and complete pedigree and provenance for every component.



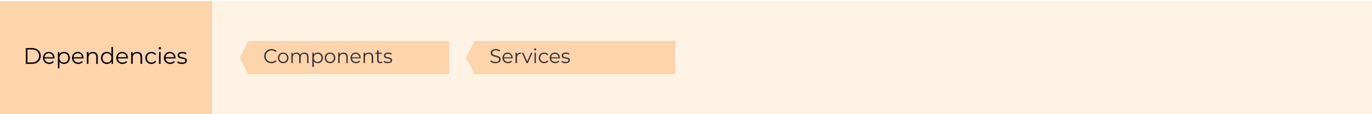
5.4.2.3 Services

Services represent external APIs that the software may call. They describe endpoint URIs, authentication requirements, and trust boundary traversals. The data flow between software and services can also be described, including the data classifications and the flow direction of each type.



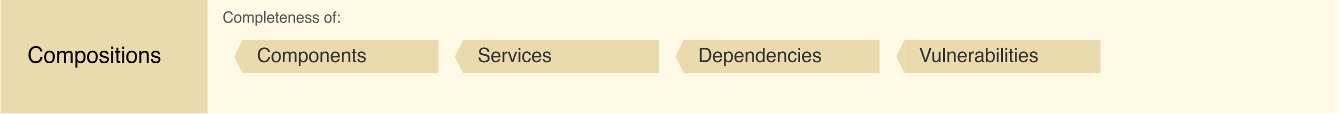
5.4.2.4 Dependencies

CycloneDX provides the ability to describe components and their dependency on other components. The dependency graph is capable of representing both direct and transitive relationships. Components that depend on services can be represented in the dependency graph, and services that depend on other services can be represented as well.



5.4.2.5 Compositions

Compositions describe constituent parts (including components, services, and dependency relationships) and their completeness. The aggregate of each composition can be described as complete, incomplete, incomplete first-party only, incomplete third-party only, or unknown.



Compositions

Completeness of:

Components

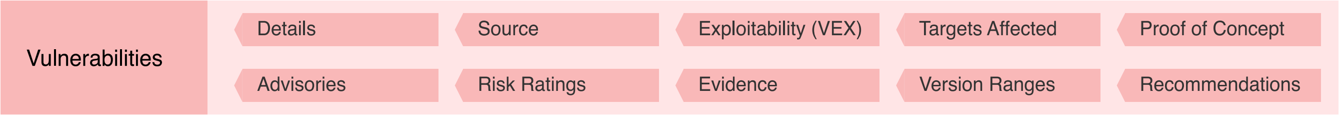
Services

Dependencies

Vulnerabilities

5.4.2.6 Vulnerabilities

Known vulnerabilities inherited from the use of third-party and open-source software and the exploitability of the vulnerabilities can be communicated with CycloneDX. Previously unknown vulnerabilities affecting both components and services may also be disclosed using CycloneDX, making it ideal for both vulnerability disclosure and VEX use cases.



Vulnerabilities

Details

Source

Exploitability (VEX)

Targets Affected

Proof of Concept

Advisories

Risk Ratings

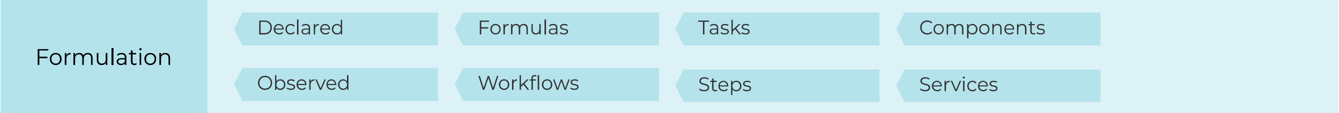
Evidence

Version Ranges

Recommendations

5.4.2.7 Formulation

Formulation describes how something was manufactured or deployed. CycloneDX achieves this through the support of multiple formulas, workflows, tasks, and steps, which represent the declared formulation for reproduction along with the observed formula describing the actions which transpired in the manufacturing process.



Formulation

Declared

Formulas

Tasks

Components

Observed

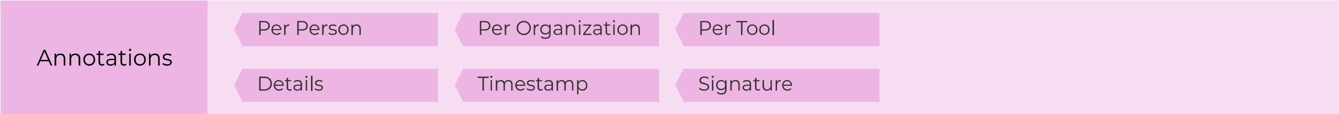
Workflows

Steps

Services

5.4.2.8 Annotations

Annotations contain comments, notes, explanations, or similar textual content which provide additional context to the object(s) being annotated. They are often automatically added to a BOM via a tool or as a result of manual review by individuals or organizations. Annotations can be independently signed and verified using digital signatures.



Annotations

Per Person

Per Organization

Per Tool

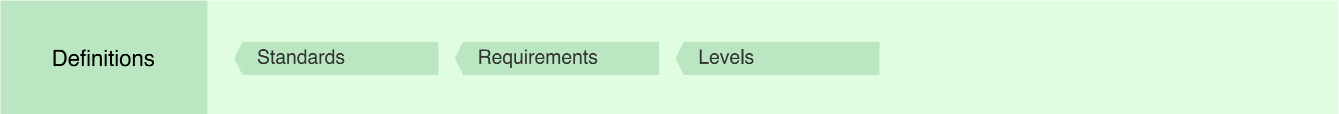
Details

Timestamp

Signature

5.4.2.9 Definitions

Standards, requirements, levels, and all supporting documentation are defined here. CycloneDX provides a general-purpose, machine-readable way to define virtually any type of standard. Security standards such as OWASP ASVS, MASVS, SCVS, and SAMM are available in CycloneDX format. Standards from other bodies are available as well. Additionally, organizations can create internal standards and represent them in CycloneDX.



Definitions

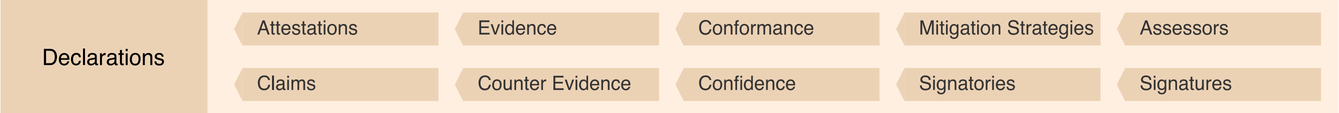
Standards

Requirements

Levels

5.4.2.10 Declarations

Declarations describe the conformance to standards. Each declaration may include attestations, claims, counter-claims, evidence, counter-evidence, along with conformance and confidence. Signatories can also be declared and supports both digital and analogue signatures. Declarations provide the basis for "compliance-as-code".



Declarations

Attestations

Evidence

Conformance

Mitigation Strategies

Assessors

Claims

Counter Evidence

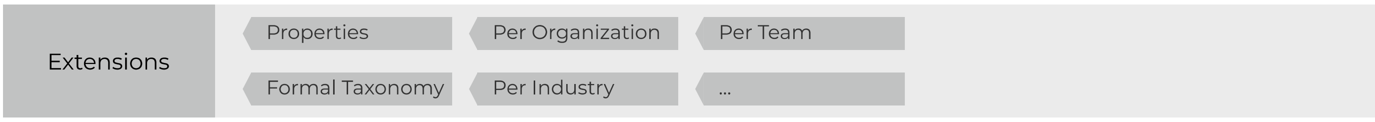
Confidence

Signatories

Signatures

5.4.2.11 Extensions

Multiple extension points exist throughout the CycloneDX object model, allowing fast prototyping of new capabilities and support for specialized and future use cases. The CycloneDX project maintains extensions that are beneficial to the larger community. The project encourages community participation and the development of extensions that target specialized or industry-specific use cases.



Extensions

Properties

Per Organization

Per Team

Formal Taxonomy

Per Industry

...

5.5 Serialization Formats

CycloneDX can be represented in JSON, XML, and Protocol Buffers (protobuf) and has corresponding schemas for each.

Format	Resource	URL
JSON	Documentation	https://cyclonedx.org/docs/latest/json/
JSON	Schema	https://cyclonedx.org/schema/bom-1.6.schema.json
XML	Documentation	https://cyclonedx.org/docs/latest/xml/
XML	Schema	https://cyclonedx.org/schema/bom-1.6.xsd
Protobuf	Schema	https://cyclonedx.org/schema/bom-1.6.proto

CycloneDX relies exclusively on JSON Schema, XML Schema, and protobuf for validation. The entirety of the specification can be validated using officially supported CycloneDX tools or via hundreds of available validators that support JSON Schema, XML Schema, or protobuf.

NOTE The CycloneDX JSON Schema is the reference implementation for the Ecma standard.

6 CycloneDX Bill of Materials Standard

Location: /

Type: Object

Table 1 – Properties for the root object

Property	Type	Requirement	Description
bomFormat	String	Required	Specifies the format of the BOM. This helps to identify the file as CycloneDX since BOMs do not have a filename convention, nor does JSON schema support namespaces. This value must be "CycloneDX".
specVersion	String	Required	The version of the CycloneDX specification the BOM conforms to.
serialNumber	String	Optional	Every BOM generated SHOULD have a unique serial number, even if the contents of the BOM have not changed over time. If specified, the serial number must conform to RFC 4122. Use of serial numbers is recommended.

Property	Type	Requirement	Description
version	Integer	Optional	Whenever an existing BOM is modified, either manually or through automated processes, the version of the BOM SHOULD be incremented by 1. When a system is presented with multiple BOMs with identical serial numbers, the system SHOULD use the most recent version of the BOM. The default version is '1'.
metadata	Object	Optional	Provides additional information about a BOM.
components	Array	Optional	A list of software and hardware components.
services	Array	Optional	A list of services. This may include microservices, function-as-a-service, and other types of network or intra-process services.
externalReferences	Array	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.
dependencies	Array	Optional	Provides the ability to document dependency relationships including provided & implemented components.
compositions	Array	Optional	Compositions describe constituent parts (including components, services, and dependency relationships) and their completeness. The completeness of vulnerabilities expressed in a BOM may also be described.
vulnerabilities	Array	Optional	Vulnerabilities identified in components or services.
annotations	Array	Optional	Comments made by people, organizations, or tools about any object with a bom-ref, such as components, services, vulnerabilities, or the BOM itself. Unlike inventory information, annotations may contain opinions or commentary from various stakeholders. Annotations may be inline (with inventory) or externalized via BOM-Link and may optionally be signed.
formulation	Array	Optional	Describes how a component or service was manufactured or deployed. This is achieved through the use of formulas, workflows, tasks, and steps, which declare the precise steps to reproduce along with the observed formulas describing the steps which transpired in the manufacturing process.
declarations	Object	Optional	The list of declarations which describe the conformance to standards. Each declaration may include attestations, claims, and evidence.
definitions	Object	Optional	A collection of reusable objects that are defined and may be used elsewhere in the BOM.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of

Property	Type	Requirement	Description
			interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.
signature	Array	Optional	Enveloped signature in JSON Signature Format (JSF).

6.1 BOM Format

Location: /bomFormat
Property: bomFormat (Required)

Type: String

Description: Specifies the format of the BOM. This helps to identify the file as CycloneDX since BOMs do not have a filename convention, nor does JSON schema support namespaces. This value must be "CycloneDX".

Enumeration: Must be one of:

- CycloneDX

6.2 CycloneDX Specification Version

Location: /specVersion
Property: specVersion (Required)

Type: String

Description: The version of the CycloneDX specification the BOM conforms to.

Examples:

- 1.6

6.3 BOM Serial Number

Location: /serialNumber
Property: serialNumber (Optional)

Type: String

Pattern Constraint: ^urn:uuid:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$

Description: Every BOM generated SHOULD have a unique serial number, even if the contents of the BOM have not changed over time. If specified, the serial number must conform to RFC 4122. Use of serial numbers is recommended.

Examples:

- urn:uuid:3e671687-395b-41f5-a30f-a58921a69b79

6.4 BOM Version

Location: /version
Property: version (Optional)

Type: Integer

Minimum Value: 1

Default Value: 1

Description: Whenever an existing BOM is modified, either manually or through automated processes, the version of the BOM SHOULD be incremented by 1. When a system is presented with multiple BOMs with identical serial numbers, the system SHOULD use the most recent version of the BOM. The default version is '1'.

Examples:

- 1

7 BOM Metadata

Location: /metadata
Property: metadata (Optional)

Type: Object

Description: Provides additional information about a BOM.

Table 2 – Properties for the metadata object

Property	Type	Requirement	Description
timestamp	String	Optional	The date and time (timestamp) when the BOM was created.
lifecycles	Array	Optional	Lifecycles communicate the stage(s) in which data in the BOM was captured. Different types of data may be available at various phases of a lifecycle, such as the Software Development Lifecycle (SDLC), IT Asset Management (ITAM), and Software Asset Management (SAM). Thus, a BOM may include data specific to or only obtainable in a given lifecycle.
tools	Array	Optional	The tool(s) used in the creation, enrichment, and validation of the BOM.
manufacturer	Object	Optional	The organization that created the BOM. Manufacturer is common in BOMs created through automated processes. BOMs created through manual means may have @.authors instead.
authors	Array	Optional	The person(s) who created the BOM. Authors are common in BOMs created through manual processes. BOMs created through automated means may have @.manufacturer instead.
component	Object	Optional	The component that the BOM describes.
manufacture	Object	Optional	[Deprecated] This will be removed in a future version. Use the @.component.manufacturer instead. The organization that manufactured the component that the BOM describes.
supplier	Object	Optional	The organization that supplied the component that the BOM describes. The supplier may often be the manufacturer, but may also be a distributor or repackager.
licenses	Array	Optional	The license information for the BOM document. This may be different from the license(s) of the component(s) that the BOM describes.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

7.1 Timestamp

Location: /metadata/timestamp
Property: timestamp (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time (timestamp) when the BOM was created.

7.2 Lifecycles

Location: /metadata/lifecycles
Property: lifecycles (Optional)

Type: Array

Description: Lifecycles communicate the stage(s) in which data in the BOM was captured. Different types of data may be available at various phases of a lifecycle, such as the Software Development Lifecycle (SDLC), IT Asset Management (ITAM), and Software Asset Management (SAM). Thus, a BOM may include data specific to or only obtainable in a given lifecycle. Each item of this array must be a Lifecycle object.

7.2.1 Lifecycle

Location: /metadata/lifecycles/[]

Type: Object
Description: The product lifecycle(s) that this BOM represents.

*Must be **one of**:*

1. Pre-Defined Phase
2. Custom Phase

7.2.2 Pre-Defined Phase

Type: Object

Table 3 – Properties for pre-defined phase

Property	Type	Requirement	Description
phase	String	Required	A pre-defined phase in the product lifecycle.

7.2.3 Custom Phase

Type: Object

Table 4 – Properties for custom phase

Property	Type	Requirement	Description
name	String	Required	The name of the lifecycle phase
description	String	Optional	The description of the lifecycle phase

7.2.4 Phase

Location: /metadata/lifecycles/[]/phase

Property: lifecycles (Required)

Type: String

Description: A pre-defined phase in the product lifecycle.

Table 5 – Enumeration of possible values

Value	Description
design	BOM produced early in the development lifecycle containing an inventory of components and services that are proposed or planned to be used. The inventory may need to be procured, retrieved, or resourced prior to use.
pre-build	BOM consisting of information obtained prior to a build process and may contain source files and development artifacts and manifests. The inventory may need to be resolved and retrieved prior to use.
build	BOM consisting of information obtained during a build process where component inventory is available for use. The precise versions of resolved components are usually available at this time as well as the provenance of where the components were retrieved from.
post-build	BOM consisting of information obtained after a build process has completed and the resulting components(s) are available for further analysis. Built components may exist as the result of a CI/CD process, may have been installed or deployed to a system or device, and may need to be retrieved or extracted from the system or device.
operations	BOM produced that represents inventory that is running and operational. This may include staging or production environments and will generally encompass multiple SBOMs describing the applications and operating system, along with HBOMs describing the hardware that makes up the system. Operations Bill of Materials (OBOM) can provide full-stack inventory of runtime environments, configurations, and additional dependencies.
discovery	BOM consisting of information observed through network discovery providing point-in-time enumeration of embedded, on-premise, and cloud-native services such as server applications, connected devices, microservices, and serverless functions.
decommission	BOM containing inventory that will be, or has been retired from operations.

7.2.5 Name

Location: /metadata/lifecycles/[]/name
Property: lifecycles (Required)

Type: String

Description: The name of the lifecycle phase

7.2.6 Description

Location: /metadata/lifecycles/[]/description
Property: lifecycles (Optional)

Type: String

Description: The description of the lifecycle phase

7.3 Tools

Location: /metadata/tools

Property: tools (Optional)

Description: The tool(s) used in the creation, enrichment, and validation of the BOM.

*Must be **one of**:*

1. Tools
2. Tools (legacy)

7.3.1 Tools

Type: Object
Description: The tool(s) used in the creation, enrichment, and validation of the BOM.

Table 6 – Properties for tools

Property	Type	Requirement	Description
components	Array	Optional	A list of software and hardware components used as tools. Refer to the component definition at /components/[]
services	Array	Optional	A list of services used as tools. This may include microservices, function-as-a-service, and other types of network or intra-process services. Refer to the service definition at /services/[]

7.3.2 Tools (legacy)

Type: Array
Description: [Deprecated] The tool(s) used in the creation, enrichment, and validation of the BOM.

7.3.3 Components

Location: /metadata/tools/components
Property: tools (Optional)

Type: Array

Description: A list of software and hardware components used as tools. Each item of this array must be a Component object.

7.3.3.1 Component

Location: /metadata/tools/components/[]

Type: Object
Reference: Refer to the component definition at /components/[]

All items must be unique.

7.3.4 Services

Location: /metadata/tools/services
Property: tools (Optional)

Type: Array

Description: A list of services used as tools. This may include microservices, function-as-a-service, and other types of network or intra-process services. Each item of this array must be a Service object.

7.3.4.1 Service

Location: /metadata/tools/services/[]

Type: Object

Reference: Refer to the service definition at /services/[]

All items must be unique.

7.3.4.1.1 Tools (legacy)

Location: /metadata/tools
Property: tools

Type: Array

Description: [Deprecated] The tool(s) used in the creation, enrichment, and validation of the BOM. Each item of this array must be a Tool object.

7.3.5 Tool

Location: /metadata/tools/[]

Type: Object

Description: [Deprecated] This will be removed in a future version. Use component or service instead. Information about the automated or manual tool used

Table 7 – Properties for the tools object

Property	Type	Requirement	Description
vendor	String	Optional	The name of the vendor who created the tool
name	String	Optional	The name of the tool
version	String	Optional	The version of the tool
hashes	Array	Optional	The hashes of the tool (if applicable).
externalReferences	Array	Optional	External references provide a way to document systems, sites, and information that may be relevant, but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

7.3.5.1 Tool Vendor

Location: /metadata/tools/[]/vendor
Property: vendor (Optional)

Type: String

Description: The name of the vendor who created the tool

7.3.5.2 Tool Name

Location: /metadata/tools/[]/name
Property: name (Optional)

Type: String

Description: The name of the tool

7.3.5.3 Tool Version

Location: /metadata/tools/[]/version
Property: version (Optional)

Type: String

Description: The version of the tool

Examples:

- 9.0.14
- v1.33.7
- 7.0.0-M1
- 2.0pre1
- 1.0.0-beta1
- 0.8.15

7.3.5.4 Hashes

Location: /metadata/tools/[]/hashes
Property: hashes (Optional)

Type: Array

Description: The hashes of the tool (if applicable). Each item of this array must be a Hash object.

7.3.5.4.1 Hash

Location: /metadata/tools/[]/hashes/[]

Type: Object

Table 8 – Properties for the hashes object

Property	Type	Requirement	Description
alg	String	Required	The algorithm that generated the hash value.
content	String	Required	The value of the hash.

7.3.5.4.2 Hash Algorithm

Location: /metadata/tools/[]/hashes/[]/alg
Property: alg (Required)

Type: String

Description: The algorithm that generated the hash value.

Enumeration: Must be one of:

- MD5
- SHA-1
- SHA-256
- SHA-384
- SHA-512
- SHA3-256
- SHA3-384
- SHA3-512
- BLAKE2b-256

- BLAKE2b-384
- BLAKE2b-512
- BLAKE3

7.3.5.4.3 Hash Value

Location: /metadata/tools/[]/hashes/[]/content
Property: content (Required)

Type: String

Pattern Constraint: ^([a-fA-F0-9]{32}|[a-fA-F0-9]{40}|[a-fA-F0-9]{64}|[a-fA-F0-9]{96}|[a-fA-F0-9]{128})$

Description: The value of the hash.

Examples:

- 3942447fac867ae5cdb3229b658f4d48

7.3.5.5 External References

Location: /metadata/tools/[]/externalReferences
Property: externalReferences (Optional)

Type: Array

Description: External references provide a way to document systems, sites, and information that may be relevant, but are not included with the BOM. They may also establish specific relationships within or external to the BOM. Each item of this array must be an External Reference object.

Reference: Refer to the external reference definition at /externalReferences/[]

7.3.5.5.1 External Reference

Location: /metadata/tools/[]/externalReferences/[]

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

7.4 BOM Manufacturer

Location: /metadata/manufacturer
Property: manufacturer (Optional)

Type: Object

Description: The organization that created the BOM. Manufacturer is common in BOMs created through automated processes. BOMs created through manual means may have @.authors instead.

Table 9 – Properties for the manufacturer object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization

Property	Type	Requirement	Description
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

7.4.1 BOM Reference

Location: /metadata/manufacturer/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

7.4.2 Organization Name

Location: /metadata/manufacturer/name
Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

7.4.3 Organization Address

Location: /metadata/manufacturer/address
Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 10 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

7.4.3.1 BOM Reference

Location: /metadata/manufacturer/address/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

7.4.3.2 Country

Location: /metadata/manufacturer/address/country
Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

7.4.3.3 Region

Location: /metadata/manufacturer/address/region
Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

7.4.3.4 Locality

Location: /metadata/manufacturer/address/locality
Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

7.4.3.5 Post Office Box Number

Location: /metadata/manufacturer/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

7.4.3.6 Postal Code

Location: /metadata/manufacturer/address/postalCode
Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

7.4.3.7 Street Address

Location: /metadata/manufacturer/address/streetAddress
Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

7.4.4 Organization URL(s)

Location: /metadata/manufacturer/url
Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

7.4.5 Organizational Contact

Location: /metadata/manufacturer/contact
Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

7.4.5.1 Organizational Contact

Location: /metadata/manufacturer/contact/[]

Type: Object

Table 11 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

7.4.5.1.1 BOM Reference

Location: /metadata/manufacturer/contact/[]/bom-ref

Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

7.4.5.1.2 Name

Location: /metadata/manufacturer/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

7.4.5.1.3 Email Address

Location: /metadata/manufacturer/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

7.4.5.1.4 Phone

Location: /metadata/manufacturer/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

7.5 BOM Authors

Location: /metadata/authors
Property: authors (Optional)

Type: Array

Description: The person(s) who created the BOM. Authors are common in BOMs created through manual processes. BOMs created through automated means may have @.manufacturer instead. Each item of this array must be an Organizational Contact object.

7.5.1 Organizational Contact

Location: /metadata/authors/[]

Type: Object

Table 12 – Properties for the authors object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

7.5.1.1 BOM Reference

Location: /metadata/authors/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

7.5.1.2 Name

Location: /metadata/authors/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

7.5.1.3 Email Address

Location: /metadata/authors/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

7.5.1.4 Phone

Location: /metadata/authors/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

7.6 Component

Location: /metadata/component
Property: component (Optional)

Type: Object

Description: The component that the BOM describes.

Reference: Refer to the component definition at /components/[]

7.7 Component Manufacture (legacy)

Location: /metadata/manufacture
Property: manufacture (Optional and Deprecated)

Type: Object

Description: [Deprecated] This will be removed in a future version. Use the @.component.manufacturer instead. The organization that manufactured the component that the BOM describes.

Table 13 – Properties for the manufacture object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

7.7.1 BOM Reference

Location: /metadata/manufacture/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

7.7.2 Organization Name

Location: /metadata/manufacture/name
Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

7.7.3 Organization Address

Location: /metadata/manufacture/address
Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 14 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

7.7.3.1 BOM Reference

Location: /metadata/manufacture/address/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

7.7.3.2 Country

Location: /metadata/manufacture/address/country
Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

7.7.3.3 Region

Location: /metadata/manufacture/address/region
Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

7.7.3.4 Locality

Location: /metadata/manufacture/address/locality
Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

7.7.3.5 Post Office Box Number

Location: /metadata/manufacture/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

7.7.3.6 Postal Code

Location: /metadata/manufacture/address/postalCode
Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

7.7.3.7 Street Address

Location: /metadata/manufacture/address/streetAddress
Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

7.7.4 Organization URL(s)

Location: /metadata/manufacture/url
Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

7.7.5 Organizational Contact

Location: /metadata/manufacture/contact

Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

7.7.5.1 Organizational Contact

Location: /metadata/manufacture/contact/[]

Type: Object

Table 15 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

7.7.5.1.1 BOM Reference

Location: /metadata/manufacture/contact/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

7.7.5.1.2 Name

Location: /metadata/manufacture/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

7.7.5.1.3 Email Address

Location: /metadata/manufacture/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

7.7.5.1.4 Phone

Location: /metadata/manufacture/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

7.8 Supplier

Location: /metadata/supplier
Property: supplier (Optional)

Type: Object

Description: The organization that supplied the component that the BOM describes. The supplier may often be the manufacturer, but may also be a distributor or repackager.

Table 16 – Properties for the supplier object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

7.8.1 BOM Reference

Location: /metadata/supplier/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

7.8.2 Organization Name

Location: /metadata/supplier/name
Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

7.8.3 Organization Address

Location: /metadata/supplier/address
Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 17 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

7.8.3.1 BOM Reference

Location: /metadata/supplier/address/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

7.8.3.2 Country

Location: /metadata/supplier/address/country
Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

7.8.3.3 Region

Location: /metadata/supplier/address/region
Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

7.8.3.4 Locality

Location: /metadata/supplier/address/locality
Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

7.8.3.5 Post Office Box Number

Location: /metadata/supplier/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

7.8.3.6 Postal Code

Location: /metadata/supplier/address/postalCode
Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

7.8.3.7 Street Address

Location: /metadata/supplier/address/streetAddress
Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

7.8.4 Organization URL(s)

Location: /metadata/supplier/url
Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

7.8.5 Organizational Contact

Location: /metadata/supplier/contact

Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

7.8.5.1 Organizational Contact

Location: /metadata/supplier/contact/[]

Type: Object

Table 18 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

7.8.5.1.1 BOM Reference

Location: /metadata/supplier/contact/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

7.8.5.1.2 Name

Location: /metadata/supplier/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

7.8.5.1.3 Email Address

Location: /metadata/supplier/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

7.8.5.1.4 Phone

Location: /metadata/supplier/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

7.9 BOM Licence(s)

Location: /metadata/licenses
Property: licenses (Optional)

Type: Array

Description: The licence information for the BOM document. This may be different from the licence(s) of the component(s) that the BOM describes.

Reference: Refer to the license definition at /components/[]/licenses

7.10 Properties

Location: /metadata/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

7.10.1 Lightweight name-value pair

Location: /metadata/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 19 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

7.10.1.1 Name

Location: /metadata/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

7.10.1.2 Value

Location: /metadata/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

8 Components

Location: /components
Property: components (Optional)

Type: Array

Description: A list of software and hardware components. Each item of this array must be a Component object.

8.1 Component

Location: /components/[]

Type: Object

Table 20 – Properties for the components object

Property	Type	Requirement	Description
type	String	Required	Specifies the type of component. For software components, classify as application if no more specific appropriate classification is available or cannot be determined for the component.
mime-type	String	Optional	The optional mime-type of the component. When used on file components, the mime-type can provide additional context about the kind of file being represented, such as an image, font, or executable. Some library or framework components may also have an associated mime-type.
bom-ref	String	Optional	An optional identifier which can be used to reference the component elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
supplier	Object	Optional	The organization that supplied the component. The supplier may often be the manufacturer, but may also be a distributor or repackager.
manufacturer	Object	Optional	The organization that created the component. Manufacturer is common in components created through automated processes. Components created through manual means may have @.authors instead.
authors	Array	Optional	The person(s) who created the component. Authors are common in components created through manual processes. Components created through automated means may have @.manufacturer instead.

Property	Type	Requirement	Description
author	String	Optional	[Deprecated] This will be removed in a future version. Use @.authors or @.manufacturer instead. The person(s) or organization(s) that authored the component
publisher	String	Optional	The person(s) or organization(s) that published the component
group	String	Optional	The grouping name or identifier. This will often be a shortened, single name of the company or project that produced the component, or the source package or domain name. Whitespace and special characters should be avoided. Examples include: apache, org.apache.commons, and apache.org.
name	String	Required	The name of the component. This will often be a shortened, single name of the component. Examples: commons-lang3 and jquery
version	String	Optional	The component version. The version should ideally comply with semantic versioning but is not enforced.
description	String	Optional	Specifies a description for the component
scope	String	Optional	Specifies the scope of the component. If scope is not specified, 'required' scope SHOULD be assumed by the consumer of the BOM.
hashes	Array	Optional	The hashes of the component.
licenses	Array	Optional	EITHER (list of SPDX licenses and/or named licenses) OR (tuple of one SPDX License Expression)
copyright	String	Optional	A copyright notice informing users of the underlying claims to copyright ownership in a published work.
cpe	String	Optional	Asserts the identity of the component using CPE. The CPE must conform to the CPE 2.2 or 2.3 specification. See https://nvd.nist.gov/products/cpe. Refer to @.evidence.identity to optionally provide evidence that substantiates the assertion of the component's identity.
purl	String	Optional	Asserts the identity of the component using package-url (purl). The purl, if specified, must be valid and conform to the specification defined at: https://github.com/package-url/purl-spec. Refer to @.evidence.identity to optionally provide evidence that substantiates the assertion of the component's identity.
omniborId	Array	Optional	Asserts the identity of the component using the OmniBOR Artifact ID. The OmniBOR, if specified, must be valid and conform to the specification defined at: https://www.iana.org/assignments/uri-schemes/prov/gitoid. Refer to @.evidence.identity to optionally provide evidence that substantiates the assertion of the component's identity.

Property	Type	Requirement	Description
swhid	Array	Optional	Asserts the identity of the component using the Software Heritage persistent identifier (SWHID). The SWHID, if specified, must be valid and conform to the specification defined at: https://docs.softwareheritage.org/devel/swh-model/persistent-identifiers.html. Refer to @.evidence.identity to optionally provide evidence that substantiates the assertion of the component's identity.
swid	Object	Optional	Asserts the identity of the component using ISO-IEC 19770-2 Software Identification (SWID) Tags. Refer to @.evidence.identity to optionally provide evidence that substantiates the assertion of the component's identity.
modified	Boolean	Optional	[Deprecated] This will be removed in a future version. Use the pedigree element instead to supply information on exactly how the component was modified. A boolean value indicating if the component has been modified from the original. A value of true indicates the component is a derivative of the original. A value of false indicates the component has not been modified from the original.
pedigree	Object	Optional	Component pedigree is a way to document complex supply chain scenarios where components are created, distributed, modified, redistributed, combined with other components, etc. Pedigree supports viewing this complex chain from the beginning, the end, or anywhere in the middle. It also provides a way to document variants where the exact relation may not be known.
externalReferences	Array	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.
components	Array	Optional	A list of software and hardware components included in the parent component. This is not a dependency tree. It provides a way to specify a hierarchical representation of component assemblies, similar to system → subsystem → parts assembly in physical supply chains.
evidence	Object	Optional	Provides the ability to document evidence collected through various forms of extraction or analysis.
releaseNotes	Object	Optional	Specifies optional release notes.
modelCard	Object	Optional	A model card describes the intended uses of a machine learning model and potential limitations, including biases and ethical considerations. Model cards typically contain the training parameters, which datasets were used to train the model, performance metrics, and other relevant data useful for ML transparency. This object SHOULD be specified for any component of type machine-learning-model and must not be specified for other component types.

Property	Type	Requirement	Description
data	Array	Optional	This object SHOULD be specified for any component of type data and must not be specified for other component types.
cryptoProperties	Object	Optional	Cryptographic assets have properties that uniquely define them and that make them actionable for further reasoning. As an example, it makes a difference if one knows the algorithm family (e.g. AES) or the specific variant or instantiation (e.g. AES-128-GCM). This is because the security level and the algorithm primitive (authenticated encryption) are only defined by the definition of the algorithm variant. The presence of a weak cryptographic algorithm like SHA1 vs. HMAC-SHA1 also makes a difference.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.
tags	Array	Optional	Textual strings that aid in discovery, search, and retrieval of the associated object. Tags often serve as a way to group or categorize similar or related objects by various attributes.
signature	Array	Optional	Enveloped signature in JSON Signature Format (JSF).

8.1.1 Component Type

Location: /components/[]/type
Property: type (Required)

Type: String

Description: Specifies the type of component. For software components, classify as application if no more specific appropriate classification is available or cannot be determined for the component.

Examples:

- library

Table 21 – Enumeration of possible values

Value	Description
application	A software application. Refer to https://en.wikipedia.org/wiki/Application_software for information about applications.
framework	A software framework. Refer to https://en.wikipedia.org/wiki/Software_framework for information on how frameworks vary slightly from libraries.
library	A software library. Refer to https://en.wikipedia.org/wiki/Library_(computing) for information about libraries. All third-party and open source reusable components will likely be a library.

Value	Description
	If the library also has key features of a framework, then it should be classified as a framework. If not, or is unknown, then specifying library is recommended.
container	A packaging and/or runtime format, not specific to any particular technology, which isolates software inside the container from software outside of a container through virtualization technology. Refer to https://en.wikipedia.org/wiki/OS-level_virtualization.
platform	A runtime environment which interprets or executes software. This may include runtimes such as those that execute bytecode or low-code/no-code application platforms.
operating-system	A software operating system without regard to deployment model (i.e. installed on physical hardware, virtual machine, image, etc) Refer to https://en.wikipedia.org/wiki/Operating_system.
device	A hardware device such as a processor or chip-set. A hardware device containing firmware SHOULD include a component for the physical hardware itself and another component of type 'firmware' or 'operating-system' (whichever is relevant), describing information about the software running on the device. See also the list of known device properties.
device-driver	A special type of software that operates or controls a particular type of device. Refer to https://en.wikipedia.org/wiki/Device_driver.
firmware	A special type of software that provides low-level control over a device's hardware. Refer to https://en.wikipedia.org/wiki/Firmware.
file	A computer file. Refer to https://en.wikipedia.org/wiki/Computer_file for information about files.
machine-learning-model	A model based on training data that can make predictions or decisions without being explicitly programmed to do so.
data	A collection of discrete values that convey information.
cryptographic-asset	A cryptographic asset including algorithms, protocols, certificates, keys, tokens, and secrets.

8.1.2 Mime-Type

Location: /components/[]/mime-type
Property: mime-type (Optional)

Type: String

Pattern Constraint: ^[-+a-z0-9.]+/[-+a-z0-9.]+$

Description: The optional mime-type of the component. When used on file components, the mime-type can provide additional context about the kind of file being represented, such as an image, font, or executable. Some library or framework components may also have an associated mime-type.

Examples:

- image/jpeg

8.1.3 BOM Reference

Location: /components/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the component elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.4 Component Supplier

Location: /components/[]/supplier
Property: supplier (Optional)

Type: Object

Description: The organization that supplied the component. The supplier may often be the manufacturer, but may also be a distributor or repackager.

Table 22 – Properties for the supplier object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

8.1.4.1 BOM Reference

Location: /components/[]/supplier/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.4.2 Organization Name

Location: /components/[]/supplier/name
Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

8.1.4.3 Organization Address

Location: /components/[]/supplier/address
Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 23 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

8.1.4.3.1 BOM Reference

Location: /components/[]/supplier/address/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.4.3.2 Country

Location: /components/[]/supplier/address/country
Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

8.1.4.3.3 Region

Location: /components/[]/supplier/address/region
Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

8.1.4.3.4 Locality

Location: /components/[]/supplier/address/locality
Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

8.1.4.3.5 Post Office Box Number

Location: /components/[]/supplier/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

8.1.4.3.6 Postal Code

Location: /components/[]/supplier/address/postalCode
Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

8.1.4.3.7 Street Address

Location: /components/[]/supplier/address/streetAddress
Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

8.1.4.4 Organization URL(s)

Location: /components/[]/supplier/url
Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

8.1.4.5 Organizational Contact

Location: /components/[]/supplier/contact

Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

8.1.4.5.1 Organizational Contact

Location: /components/[]/supplier/contact/[]

Type: Object

Table 24 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

8.1.4.5.2 BOM Reference

Location: /components/[]/supplier/contact/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.4.5.3 Name

Location: /components/[]/supplier/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

8.1.4.5.4 Email Address

Location: /components/[]/supplier/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

8.1.4.5.5 Phone

Location: /components/[]/supplier/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

8.1.5 Component Manufacturer

Location: /components/[]/manufacturer
Property: manufacturer (Optional)

Type: Object

Description: The organization that created the component. Manufacturer is common in components created through automated processes. Components created through manual means may have @.authors instead.

Table 25 – Properties for the manufacturer object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

8.1.5.1 BOM Reference

Location: /components/[]/manufacturer/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.5.2 Organization Name

Location: /components/[]/manufacturer/name
Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

8.1.5.3 Organization Address

Location: /components/[]/manufacturer/address
Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 26 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

8.1.5.3.1 BOM Reference

Location: /components/[]/manufacturer/address/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.5.3.2 Country

Location: /components/[]/manufacturer/address/country
Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

8.1.5.3.3 Region

Location: /components/[]/manufacturer/address/region
Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

8.1.5.3.4 Locality

Location: /components/[]/manufacturer/address/locality
Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

8.1.5.3.5 Post Office Box Number

Location: /components/[]/manufacturer/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

8.1.5.3.6 Postal Code

Location: /components/[]/manufacturer/address/postalCode
Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

8.1.5.3.7 Street Address

Location: /components/[]/manufacturer/address/streetAddress
Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

8.1.5.4 Organization URL(s)

Location: /components/[]/manufacturer/url
Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

8.1.5.5 Organizational Contact

Location: /components/[]/manufacturer/contact

Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

8.1.5.5.1 Organizational Contact

Location: /components/[]/manufacturer/contact/[]

Type: Object

Table 27 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

8.1.5.5.2 BOM Reference

Location: /components/[]/manufacturer/contact/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.5.5.3 Name

Location: /components/[]/manufacturer/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

8.1.5.5.4 Email Address

Location: /components/[]/manufacturer/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

8.1.5.5.5 Phone

Location: /components/[]/manufacturer/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

8.1.6 Component Authors

Location: /components/[]/authors
Property: authors (Optional)

Type: Array

Description: The person(s) who created the component. Authors are common in components created through manual processes. Components created through automated means may have @.manufacturer instead. Each item of this array must be an Organizational Contact object.

8.1.6.1 Organizational Contact

Location: /components/[]/authors/[]

Type: Object

Table 28 – Properties for the authors object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

8.1.6.1.1 BOM Reference

Location: /components/[]/authors/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.6.1.2 Name

Location: /components/[]/authors/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

8.1.6.1.3 Email Address

Location: /components/[]/authors/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

8.1.6.1.4 Phone

Location: /components/[]/authors/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

8.1.7 Component Author (legacy)

Location: /components/[]/author
Property: author (Optional and Deprecated)

Type: String

Description: [Deprecated] This will be removed in a future version. Use @.authors or @.manufacturer instead. The person(s) or organization(s) that authored the component

Examples:

- Acme Inc

8.1.8 Component Publisher

Location: /components/[]/publisher
Property: publisher (Optional)

Type: String

Description: The person(s) or organization(s) that published the component

Examples:

- Acme Inc

8.1.9 Component Group

Location: /components/[]/group
Property: group (Optional)

Type: String

Description: The grouping name or identifier. This will often be a shortened, single name of the company or project that produced the component, or the source package or domain name. Whitespace and special characters should be avoided. Examples include: apache, org.apache.commons, and apache.org.

Examples:

- com.acme

8.1.10 Component Name

Location: /components/[]/name
Property: name (Required)

Type: String

Description: The name of the component. This will often be a shortened, single name of the component. Examples: commons-lang3 and jquery

Examples:

- tomcat-catalina

8.1.11 Component Version

Location: /components/[]/version
Property: version (Optional)

Type: String

Description: The component version. The version should ideally comply with semantic versioning but is not enforced.

Examples:

- 9.0.14
- v1.33.7
- 7.0.0-M1
- 2.0pre1
- 1.0.0-beta1
- 0.8.15

8.1.12 Component Description

Location: /components/[]/description
Property: description (Optional)

Type: String

Description: Specifies a description for the component

8.1.13 Component Scope

Location: /components/[]/scope
Property: scope (Optional)

Type: String

Default Value: required

Description: Specifies the scope of the component. If scope is not specified, 'required' scope SHOULD be assumed by the consumer of the BOM.

Table 29 – Enumeration of possible values

Value	Description
required	The component is required for runtime
optional	The component is optional at runtime. Optional components are components that are not capable of being called due to them not being installed or otherwise accessible by any means. Components that are installed but due to configuration or other restrictions are prohibited from being called must be scoped as 'required'.
excluded	Components that are excluded provide the ability to document component usage for test and other non-runtime purposes. Excluded components are not reachable within a call graph at runtime.

8.1.14 Component Hashes

Location: /components/[]/hashes
Property: hashes (Optional)

Type: Array

Description: The hashes of the component. Each item of this array must be a Hash object.

8.1.14.1 Hash

Location: /components/[]/hashes/[]

Type: Object

Table 30 – Properties for the hashes object

Property	Type	Requirement	Description
alg	String	Required	The algorithm that generated the hash value.
content	String	Required	The value of the hash.

8.1.14.1.1 Hash Algorithm

Location: /components/[]/hashes/[]/alg
Property: alg (Required)

Type: String

Description: The algorithm that generated the hash value.

Enumeration: Must be one of:

- MD5
- SHA-1
- SHA-256
- SHA-384
- SHA-512
- SHA3-256
- SHA3-384
- SHA3-512

- BLAKE2b-256
- BLAKE2b-384
- BLAKE2b-512
- BLAKE3

8.1.14.1.2 Hash Value

Location: /components/[]/hashes/[]/content
Property: content (Required)

Type: String

Pattern Constraint: ^([a-fA-F0-9]{32}|[a-fA-F0-9]{40}|[a-fA-F0-9]{64}|[a-fA-F0-9]{96}|[a-fA-F0-9]{128})$

Description: The value of the hash.

Examples:

- 3942447fac867ae5cdb3229b658f4d48

8.1.15 Component Licence(s)

Location: /components/[]/licenses
Property: licenses (Optional)

Type: Array

Description: EITHER (list of SPDX licences and/or named licences) OR (tuple of one SPDX Licence Expression)

*Must be **one of**:*

1. Multiple licenses
2. SPDX License Expression

8.1.16 Multiple licences

Type: Array
Description: A list of SPDX licences and/or named licences.

8.1.17 SPDX Licence Expression

Type: Array
Description: A tuple of exactly one SPDX Licence Expression.

8.1.18 Multiple licences

Location: /components/[]/licenses
Property: licenses

Type: Array

Description: A list of SPDX licences and/or named licences. Each item of this array must be a Licence object.

8.1.18.1 Licence

Location: /components/[]/licenses/[]

Type: Object

Table 31 – Properties for the licenses object

Property	Type	Requirement	Description
license	Array	Required	Specifies the details and attributes related to a software license. It can either include a valid SPDX license identifier or a named license, along with additional properties such as license acknowledgment, comprehensive commercial licensing information, and the full text of the license.

8.1.18.1.1 Licence

Location: /components/[]/licenses/[]/license
Property: license (Required)

Type: Object

Description: Specifies the details and attributes related to a software licence. It can either include a valid SPDX licence identifier or a named licence, along with additional properties such as licence acknowledgment, comprehensive commercial licencing information, and the full text of the licence.

Table 32 – Properties for the license object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the license elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
id	String	Optional	A valid SPDX license identifier. If specified, this value must be one of the enumeration of valid SPDX license identifiers defined in the spdx.schema.json (or spdx.xml) subschema which is synchronized with the official SPDX license list.
name	String	Optional	The name of the license. This may include the name of a commercial or proprietary license or an open source license that may not be defined by SPDX.
acknowledgement	String	Optional	Declared licenses and concluded licenses represent two different stages in the licensing process within software development. Declared licenses refer to the initial intention of the software authors regarding the licensing terms under which their code is released. On the other hand, concluded licenses are the result of a comprehensive analysis of the project's codebase to identify and confirm the actual licenses of the components used, which may differ from the initially declared licenses. While declared licenses provide an upfront indication of the licensing intentions, concluded licenses offer a more thorough understanding of the actual licensing within a project, facilitating proper compliance and risk management. Observed licenses are defined in @.evidence.licenses. Observed licenses form the evidence necessary to substantiate a concluded license.
text	Object	Optional	An optional way to include the textual content of a license.
url	String	Optional	The URL to the license file. If specified, a 'license' externalReference should also be specified for completeness

Property	Type	Requirement	Description
licensing	Object	Optional	Licensing details describing the licensor/licensee, license type, renewal and expiration dates, and other important metadata
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

8.1.18.1.2 BOM Reference

Location: /components/[]/licenses/[]/license/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the licence elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.18.1.3 Licence ID (SPDX)

Location: /components/[]/licenses/[]/license/id
Property: id (Optional)

Type: String

Description: A valid SPDX licence identifier. If specified, this value must be one of the enumeration of valid SPDX licence identifiers defined in the spdx.schema.json (or spdx.xml) subschema which is synchronized with the official SPDX licence list.

Examples:

- Apache-2.0

8.1.18.1.4 Licence Name

Location: /components/[]/licenses/[]/license/name
Property: name (Optional)

Type: String

Description: The name of the licence. This may include the name of a commercial or proprietary licence or an open source licence that may not be defined by SPDX.

Examples:

- Acme Software License

8.1.18.1.5 Licence Acknowledgement

Location: /components/[]/licenses/[]/license/acknowledgement
Property: acknowledgement (Optional)

Type: String

Description: Declared licences and concluded licences represent two different stages in the licencing process within software development. Declared licences refer to the initial intention of the software authors regarding the licencing terms under which their code is released. On the other hand, concluded licences are the result of a comprehensive analysis of the project's codebase to identify and confirm the actual licences of the components used, which may differ from the initially declared licences. While declared licences provide an upfront indication of the licencing intentions, concluded licences offer a more thorough understanding of the actual licencing within a project, facilitating proper compliance and risk management. Observed licences are defined in @.evidence.licenses. Observed licences form the evidence necessary to substantiate a concluded licence.

Table 33 – Enumeration of possible values

Value	Description
declared	Declared licences represent the initial intentions of authors regarding the licencing terms of their code.
concluded	Concluded licences are verified and confirmed.

8.1.18.1.6 Licence text

Location: /components/[]/licenses/[]/license/text
Property: text (Optional)

Type: Object

Description: An optional way to include the textual content of a licence.

Table 34 – Properties for the text object

Property	Type	Requirement	Description
contentType	String	Optional	Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.
encoding	String	Optional	Specifies the optional encoding the text is represented in.
content	String	Required	The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

8.1.18.1.7 Content-Type

Location: /components/[]/licenses/[]/license/text/contentType
Property: contentType (Optional)

Type: String

Default Value: text/plain

Description: Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.

Examples:

- text/plain
- application/json
- image/png

8.1.18.1.8 Encoding

Location: /components/[]/licenses/[]/license/text/encoding
Property: encoding (Optional)

Type: String

Description: Specifies the optional encoding the text is represented in.

Table 35 – Enumeration of possible values

Value	Description
base64	Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string.

8.1.18.1.9 Attachment Text

Location: /components/[]/licenses/[]/license/text/content
Property: content (Required)

Type: String

Description: The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

8.1.18.1.10 Licence URL

Location: /components/[]/licenses/[]/license/url
Property: url (Optional)

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL to the licence file. If specified, a 'licence' externalReference should also be specified for completeness

Examples:

- https://www.apache.org/licenses/LICENSE-2.0.txt

8.1.18.1.11 Licencing information

Location: /components/[]/licenses/[]/license/licensing
Property: licensing (Optional)

Type: Object

Description: Licencing details describing the licensor/licensee, licence type, renewal and expiration dates, and other important metadata

Table 36 – Properties for the licensing object

Property	Type	Requirement	Description
altIds	Array	Optional	License identifiers that may be used to manage licenses and their lifecycle
licensor	Array	Optional	The individual or organization that grants a license to another individual or organization
licensee	Array	Optional	The individual or organization for which a license was granted to
purchaser	Array	Optional	The individual or organization that purchased the license

Property	Type	Requirement	Description
purchaseOrder	String	Optional	The purchase order identifier the purchaser sent to a supplier or vendor to authorize a purchase
licenseTypes	Array	Optional	The type of license(s) that was granted to the licensee.
lastRenewal	String	Optional	The timestamp indicating when the license was last renewed. For new purchases, this is often the purchase or acquisition date. For non-perpetual licenses or subscriptions, this is the timestamp of when the license was last renewed.
expiration	String	Optional	The timestamp indicating when the current license expires (if applicable).

8.1.18.1.12 Alternate Licence Identifiers

Location: /components/[]/licenses/[]/license/licensing/altIds
Property: altIds (Optional)

Type: array (of String)

Description: Licence identifiers that may be used to manage licences and their lifecycle Each item of this array must be a string.

8.1.18.1.13 Licensor

Location: /components/[]/licenses/[]/license/licensing/licensor
Property: licensor (Optional)

Type: Object

Description: The individual or organization that grants a licence to another individual or organization

Table 37 – Properties for the licensor object

Property	Type	Requirement	Description
organization	Object	Optional	The organization that granted the license
individual	Object	Optional	The individual, not associated with an organization, that granted the license

8.1.18.1.14 Licensor (Organization)

Location: /components/[]/licenses/[]/license/licensing/licensor/organization
Property: organization (Optional)

Type: Object

Description: The organization that granted the licence

Table 38 – Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

8.1.18.1.15 BOM Reference

Location: /components/[]/licenses/[]/license/licensing/licensor/organization/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.18.1.16 Organization Name

Location: /components/[]/licenses/[]/license/licensing/licensor/organization/name
Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

8.1.18.1.17 Organization Address

Location: /components/[]/licenses/[]/license/licensing/licensor/organization/address
Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 39 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.

Property	Type	Requirement	Description
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

8.1.18.1.18 BOM Reference

Location: /components/[]/licenses/[]/license/licensing/licensor/organization/address/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.18.1.19 Country

Location: /components/[]/licenses/[]/license/licensing/licensor/organization/address/country
Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

8.1.18.1.20 Region

Location: /components/[]/licenses/[]/license/licensing/licensor/organization/address/region
Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

8.1.18.1.21 Locality

Location: /components/[]/licenses/[]/license/licensing/licensor/organization/address/locality
Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

8.1.18.1.22 Post Office Box Number

Location: /components/[]/licenses/[]/license/licensing/licensor/organization/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

8.1.18.1.23 Postal Code

Location: /components/[]/licenses/[]/license/licensing/licensor/organization/address/postalCode
Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

8.1.18.1.24 Street Address

Location: /components/[]/licenses/[]/license/licensing/licensor/organization/address/streetAddress
Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

8.1.18.1.25 Organization URL(s)

Location: /components/[]/licenses/[]/license/licensing/licensor/organization/url
Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

8.1.18.1.26 Organizational Contact

Location: /components/[]/licenses/[]/license/licensing/licensor/organization/contact
Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

8.1.18.1.27 Organizational Contact

Location: /components/[]/licenses/[]/license/licensing/licensor/organization/contact/[]

Type: Object

Table 40 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

8.1.18.1.28 BOM Reference

Location: /components/[]/licenses/[]/license/licensing/licensor/organization/contact/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.18.1.29 Name

Location: /components/[]/licenses/[]/license/licensing/licensor/organization/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

8.1.18.1.30 Email Address

Location: /components/[]/licenses/[]/license/licensing/licensor/organization/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

8.1.18.1.31 Phone

Location: /components/[]/licenses/[]/license/licensing/licensor/organization/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

8.1.18.1.32 Organizational Contact

Location: /components/[]/licenses/[]/license/licensing/licensor/individual
Property: individual (Optional)

Type: Object

Description: The individual, not associated with an organization, that granted the licence

Table 41 – Properties for the individual object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

8.1.18.1.33 BOM Reference

Location: /components/[]/licenses/[]/license/licensing/licensor/individual/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.18.1.34 Name

Location: /components/[]/licenses/[]/license/licensing/licensor/individual/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

8.1.18.1.35 Email Address

Location: /components/[]/licenses/[]/license/licensing/licensor/individual/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

8.1.18.1.36 Phone

Location: /components/[]/licenses/[]/license/licensing/licensor/individual/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

8.1.18.1.37 Licensee

Location: /components/[]/licenses/[]/license/licensing/licensee
Property: licensee (Optional)

Type: Object

Description: The individual or organization for which a licence was granted to

Table 42 – Properties for the licensee object

Property	Type	Requirement	Description
organization	Object	Optional	The organization that was granted the license
individual	Object	Optional	The individual, not associated with an organization, that was granted the license

8.1.18.1.38 Licensee (Organization)

Location: /components/[]/licenses/[]/license/licensing/licensee/organization
Property: organization (Optional)

Type: Object

Description: The organization that was granted the licence

Table 43 – Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

8.1.18.1.39 BOM Reference

Location: /components/[]/licenses/[]/license/licensing/licensee/organization/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.18.1.40 Organization Name

Location: /components/[]/licenses/[]/license/licensing/licensee/organization/name
Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

8.1.18.1.41 Organization Address

Location: /components/[]/licenses/[]/license/licensing/licensee/organization/address
Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 44 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

8.1.18.1.42 BOM Reference

Location: /components/[]/licenses/[]/license/licensing/licensee/organization/address/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.18.1.43 Country

Location: /components/[]/licenses/[]/license/licensing/licensee/organization/address/country
Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

8.1.18.1.44 Region

Location: /components/[]/licenses/[]/license/licensing/licensee/organization/address/region
Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

8.1.18.1.45 Locality

Location: /components/[]/licenses/[]/license/licensing/licensee/organization/address/locality
Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

8.1.18.1.46 Post Office Box Number

Location: /components/[]/licenses/[]/license/licensing/licensee/organization/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

8.1.18.1.47 Postal Code

Location: /components/[]/licenses/[]/license/licensing/licensee/organization/address/postalCode
Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

8.1.18.1.48 Street Address

Location: /components/[]/licenses/[]/license/licensing/licensee/organization/address/streetAddress
Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

8.1.18.1.49 Organization URL(s)

Location: /components/[]/licenses/[]/license/licensing/licensee/organization/url
Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

8.1.18.1.50 Organizational Contact

Location: /components/[]/licenses/[]/license/licensing/licensee/organization/contact
Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

8.1.18.1.51 Organizational Contact

Location: /components/[]/licenses/[]/license/licensing/licensee/organization/contact/[]

Type: Object

Table 45 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

8.1.18.1.52 BOM Reference

Location: /components/[]/licenses/[]/license/licensing/licensee/organization/contact/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.18.1.53 Name

Location: /components/[]/licenses/[]/license/licensing/licensee/organization/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

8.1.18.1.54 Email Address

Location: /components/[]/licenses/[]/license/licensing/licensee/organization/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

8.1.18.1.55 Phone

Location: /components/[]/licenses/[]/license/licensing/licensee/organization/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

8.1.18.1.56 Organizational Contact

Location: /components/[]/licenses/[]/license/licensing/licensee/individual
Property: individual (Optional)

Type: Object

Description: The individual, not associated with an organization, that was granted the licence

Table 46 – Properties for the individual object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

8.1.18.1.57 BOM Reference

Location: /components/[]/licenses/[]/license/licensing/licensee/individual/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.18.1.58 Name

Location: /components/[]/licenses/[]/license/licensing/licensee/individual/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

8.1.18.1.59 Email Address

Location: /components/[]/licenses/[]/license/licensing/licensee/individual/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

8.1.18.1.60 Phone

Location: /components/[]/licenses/[]/license/licensing/licensee/individual/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

8.1.18.1.61 Purchaser

Location: /components/[]/licenses/[]/license/licensing/purchaser
Property: purchaser (Optional)

Type: Object

Description: The individual or organization that purchased the licence

Table 47 – Properties for the purchaser object

Property	Type	Requirement	Description
organization	Object	Optional	The organization that purchased the license
individual	Object	Optional	The individual, not associated with an organization, that purchased the license

8.1.18.1.62 Purchaser (Organization)

Location: /components/[]/licenses/[]/license/licensing/purchaser/organization
Property: organization (Optional)

Type: Object

Description: The organization that purchased the licence

Table 48 – Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

8.1.18.1.63 BOM Reference

Location: /components/[]/licenses/[]/license/licensing/purchaser/organization/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.18.1.64 Organization Name

Location: /components/[]/licenses/[]/license/licensing/purchaser/organization/name
Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

8.1.18.1.65 Organization Address

Location: /components/[]/licenses/[]/license/licensing/purchaser/organization/address
Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 49 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

8.1.18.1.66 BOM Reference

Location: /components/[]/licenses/[]/license/licensing/purchaser/organization/address/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.18.1.67 Country

Location: /components/[]/licenses/[]/license/licensing/purchaser/organization/address/country
Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

8.1.18.1.68 Region

Location: /components/[]/licenses/[]/license/licensing/purchaser/organization/address/region
Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

8.1.18.1.69 Locality

Location: /components/[]/licenses/[]/license/licensing/purchaser/organization/address/locality
Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

8.1.18.1.70 Post Office Box Number

Location: /components/[]/licenses/[]/license/licensing/purchaser/organization/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

8.1.18.1.71 Postal Code

Location: /components/[]/licenses/[]/license/licensing/purchaser/organization/address/postalCode
Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

8.1.18.1.72 Street Address

Location: /components/[]/licenses/[]/license/licensing/purchaser/organization/address/streetAddress
Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

8.1.18.1.73 Organization URL(s)

Location: /components/[]/licenses/[]/license/licensing/purchaser/organization/url
Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

8.1.18.1.74 Organizational Contact

Location: /components/[]/licenses/[]/license/licensing/purchaser/organization/contact
Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

8.1.18.1.75 Organizational Contact

Location: /components/[]/licenses/[]/license/licensing/purchaser/organization/contact/[]

Type: Object

Table 50 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

8.1.18.1.76 BOM Reference

Location: /components/[]/licenses/[]/license/licensing/purchaser/organization/contact/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.18.1.77 Name

Location: /components/[]/licenses/[]/license/licensing/purchaser/organization/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

8.1.18.1.78 Email Address

Location: /components/[]/licenses/[]/license/licensing/purchaser/organization/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

8.1.18.1.79 Phone

Location: /components/[]/licenses/[]/license/licensing/purchaser/organization/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

8.1.18.1.80 Organizational Contact

Location: /components/[]/licenses/[]/license/licensing/purchaser/individual
Property: individual (Optional)

Type: Object

Description: The individual, not associated with an organization, that purchased the licence

Table 51 – Properties for the individual object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

8.1.18.1.81 BOM Reference

Location: /components/[]/licenses/[]/license/licensing/purchaser/individual/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.18.1.82 Name

Location: /components/[]/licenses/[]/license/licensing/purchaser/individual/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

8.1.18.1.83 Email Address

Location: /components/[]/licenses/[]/license/licensing/purchaser/individual/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

8.1.18.1.84 Phone

Location: /components/[]/licenses/[]/license/licensing/purchaser/individual/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

8.1.18.1.85 Purchase Order

Location: /components/[]/licenses/[]/license/licensing/purchaseOrder
Property: purchaseOrder (Optional)

Type: String

Description: The purchase order identifier the purchaser sent to a supplier or vendor to authorize a purchase

8.1.18.1.86 Licence Type

Location: /components/[]/licenses/[]/license/licensing/licenseTypes
Property: licenseTypes (Optional)

Type: array (of String)

Description: The type of licence(s) that was granted to the licensee. Each item of this array must be a string.

Table 52 – Enumeration of possible values

Value	Description
academic	A licence that grants use of software solely for the purpose of education or research.
appliance	A licence covering use of software embedded in a specific piece of hardware.
client-access	A Client Access Licence (CAL) allows client computers to access services provided by server software.
concurrent-user	A Concurrent User licence (aka floating licence) limits the number of licences for a software application and licences are shared among a larger number of users.
core-points	A licence where the core of a computer's processor is assigned a specific number of points.
custom-metric	A licence for which consumption is measured by non-standard metrics.
device	A licence that covers a defined number of installations on computers and other types of devices.
evaluation	A licence that grants permission to instal and use software for trial purposes.
named-user	A licence that grants access to the software to one or more pre-defined users.
node-locked	A licence that grants access to the software on one or more pre-defined computers or devices.
oem	An Original Equipment Manufacturer licence that is delivered with hardware, cannot be transferred to other hardware, and is valid for the life of the hardware.

Value	Description
perpetual	A licence where the software is sold on a one-time basis and the licensee can use a copy of the software indefinitely.
processor-points	A licence where each installation consumes points per processor.
subscription	A licence where the licensee pays a fee to use the software or service.
user	A licence that grants access to the software or service by a specified number of users.
other	Another licence type.

8.1.18.1.87 Last Renewal

Location: /components/[]/licenses/[]/license/licensing/lastRenewal
Property: lastRenewal (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The timestamp indicating when the licence was last renewed. For new purchases, this is often the purchase or acquisition date. For non-perpetual licences or subscriptions, this is the timestamp of when the licence was last renewed.

8.1.18.1.88 Expiration

Location: /components/[]/licenses/[]/license/licensing/expiration
Property: expiration (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The timestamp indicating when the current licence expires (if applicable).

8.1.18.1.89 Properties

Location: /components/[]/licenses/[]/license/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

8.1.18.1.90 Lightweight name-value pair

Location: /components/[]/licenses/[]/license/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different

values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 53 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

8.1.18.1.91 Name

Location: /components/[]/licenses/[]/license/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

8.1.18.1.92 Value

Location: /components/[]/licenses/[]/license/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

8.1.19 SPDX Licence Expression

Location: /components/[]/licenses
Property: licenses

Type: Array

Description: A tuple of exactly one SPDX Licence Expression.

Location: /components/[]/licenses/[]

8.1.20 Component Copyright

Location: /components/[]/copyright
Property: copyright (Optional)

Type: String

Description: A copyright notice informing users of the underlying claims to copyright ownership in a published work.

Examples:

- Acme Inc

8.1.21 Common Platform Enumeration (CPE)

Location: /components/[]/cpe
Property: cpe (Optional)

Type: String

Description: Asserts the identity of the component using CPE. The CPE must conform to the CPE 2.2 or 2.3 specification. See https://nvd.nist.gov/products/cpe. Refer to @.evidence.identity to optionally provide evidence that substantiates the assertion of the component's identity.

Examples:

- cpe:2.3:a:acme:component_framework:-:*:*:*:*:*:*:*

8.1.22 Package URL (purl)

Location: /components/[]/purl
Property: purl (Optional)

Type: String

Description: Asserts the identity of the component using package-url (purl). The purl, if specified, must be valid and conform to the specification defined at: https://github.com/package-url/purl-spec. Refer to @.evidence.identity to optionally provide evidence that substantiates the assertion of the component's identity.

Examples:

- pkg:maven/com.acme/tomcat-catalina@9.0.14?packaging=jar

8.1.23 OmniBOR Artefact Identifier (gitoid)

Location: /components/[]/omniborId
Property: omniborId (Optional)

Type: array (of String)

Description: Asserts the identity of the component using the OmniBOR Artefact ID. The OmniBOR, if specified, must be valid and conform to the specification defined at: https://www.iana.org/assignments/uri-schemes/prov/gitoid. Refer to @.evidence.identity to optionally provide evidence that substantiates the assertion of the component's identity. Each item of this array must be a string.

Examples:

- gitoid:blob:sha1:a94a8fe5ccb19ba61c4c0873d391e987982fbbd3
- gitoid:blob:sha256:9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08

8.1.24 Software Heritage Identifier

Location: /components/[]/swhid
Property: swhid (Optional)

Type: array (of String)

Description: Asserts the identity of the component using the Software Heritage persistent identifier (SWHID). The SWHID, if specified, must be valid and conform to the specification defined at: https://docs.softwareheritage.org/devel/swh-model/persistent-identifiers.html. Refer to @.evidence.identity to optionally provide evidence that substantiates the assertion of the component's identity. Each item of this array must be a string.

Examples:

- swh:1:cnt:94a9ed024d3859793618152ea559a168bbcbb5e2

8.1.25 SWID Tag

Location: /components/[]/swid
Property: swid (Optional)

Type: Object

Description: Asserts the identity of the component using ISO-IEC 19770-2 Software Identification (SWID) Tags. Refer to @.evidence.identity to optionally provide evidence that substantiates the assertion of the component's identity.

Table 54 – Properties for the swid object

Property	Type	Requirement	Description
tagId	String	Required	Maps to the tagId of a SoftwareIdentity.
name	String	Required	Maps to the name of a SoftwareIdentity.
version	String	Optional	Maps to the version of a SoftwareIdentity.
tagVersion	Integer	Optional	Maps to the tagVersion of a SoftwareIdentity.
patch	Boolean	Optional	Maps to the patch of a SoftwareIdentity.
text	Object	Optional	Specifies the metadata and content of the SWID tag.
url	String	Optional	The URL to the SWID file.

8.1.25.1 Tag ID

Location: /components/[]/swid/tagId
Property: tagId (Required)

Type: String

Description: Maps to the tagId of a SoftwareIdentity.

8.1.25.2 Name

Location: /components/[]/swid/name
Property: name (Required)

Type: String

Description: Maps to the name of a SoftwareIdentity.

8.1.25.3 Version

Location: /components/[]/swid/version
Property: version (Optional)

Type: String

Default Value: 0.0

Description: Maps to the version of a SoftwareIdentity.

8.1.25.4 Tag Version

Location: /components/[]/swid/tagVersion
Property: tagVersion (Optional)

Type: Integer

Description: Maps to the tagVersion of a SoftwareIdentity.

8.1.25.5 Patch

Location: /components/[]/swid/patch
Property: patch (Optional)

Type: Boolean

Description: Maps to the patch of a SoftwareIdentity.

8.1.25.6 Attachment text

Location: /components/[]/swid/text
Property: text (Optional)

Type: Object

Description: Specifies the metadata and content of the SWID tag.

Table 55 – Properties for the text object

Property	Type	Requirement	Description
contentType	String	Optional	Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.
encoding	String	Optional	Specifies the optional encoding the text is represented in.
content	String	Required	The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

8.1.25.6.1 Content-Type

Location: /components/[]/swid/text/contentType
Property: contentType (Optional)

Type: String

Default Value: text/plain

Description: Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.

Examples:

- text/plain
- application/json
- image/png

8.1.25.6.2 Encoding

Location: /components/[]/swid/text/encoding
Property: encoding (Optional)

Type: String

Description: Specifies the optional encoding the text is represented in.

Table 56 – Enumeration of possible values

Value	Description
base64	Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string.

8.1.25.6.3 Attachment Text

Location: /components/[]/swid/text/content
Property: content (Required)

Type: String

Description: The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

8.1.25.7 URL

Location: /components/[]/swid/url
Property: url (Optional)

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL to the SWID file.

8.1.26 Component Modified From Original

Location: /components/[]/modified
Property: modified (Optional and Deprecated)

Type: Boolean

Description: [Deprecated] This will be removed in a future version. Use the pedigree element instead to supply information on exactly how the component was modified. A boolean value indicating if the component has been modified from the original. A value of true indicates the component is a derivative of the original. A value of false indicates the component has not been modified from the original.

8.1.27 Component Pedigree

Location: /components/[]/pedigree
Property: pedigree (Optional)

Type: Object

Description: Component pedigree is a way to document complex supply chain scenarios where components are created, distributed, modified, redistributed, combined with other components, etc. Pedigree supports viewing this complex chain from the beginning, the end, or anywhere in the middle. It also provides a way to document variants where the exact relation may not be known.

Table 57 – Properties for the pedigree object

Property	Type	Requirement	Description
ancestors	Array	Optional	Describes zero or more components in which a component is derived from. This is commonly used to describe forks from existing projects where the forked version contains a ancestor node containing the original component it was forked from. For example, Component A is the original component. Component B is the component being used and documented in the BOM. However, Component B contains a pedigree node with a single ancestor documenting Component A - the original component from which Component B is derived from.
descendants	Array	Optional	Descendants are the exact opposite of ancestors. This provides a way to document all forks (and their forks) of an original or root component.
variants	Array	Optional	Variants describe relations where the relationship between the components is not known. For example, if Component A contains nearly identical code to Component B. They are both related, but it is unclear if one is derived from the other, or if they share a common ancestor.
commits	Array	Optional	A list of zero or more commits which provide a trail describing how the component deviates from an ancestor, descendant, or variant.
patches	Array	Optional	>A list of zero or more patches describing how the component deviates from an ancestor, descendant, or variant. Patches may be complementary to commits or may be used in place of commits.
notes	String	Optional	Notes, observations, and other non-structured commentary describing the components pedigree.

8.1.27.1 Ancestors

Location: /components/[]/pedigree/ancestors
Property: ancestors (Optional)

Type: Array

Description: Describes zero or more components in which a component is derived from. This is commonly used to describe forks from existing projects where the forked version contains a ancestor node containing the original component it was forked from. For example, Component A is the original component. Component B is the component being used and documented in the BOM. However, Component B contains a pedigree node with a single ancestor documenting Component A - the original component from which Component B is derived from. Each item of this array must be a Component object.

8.1.27.1.1 Component

Location: /components/[]/pedigree/ancestors/[]

Type: Object
Reference: Refer to the component definition at /components/[]

8.1.27.2 Descendants

Location: /components/[]/pedigree/descendants
Property: descendants (Optional)

Type: Array

Description: Descendants are the exact opposite of ancestors. This provides a way to document all forks (and their forks) of an original or root component. Each item of this array must be a Component object.

8.1.27.2.1 Component

Location: /components/[]/pedigree/descendants/[]

Type: Object

Reference: Refer to the component definition at /components/[]

8.1.27.3 Variants

Location: /components/[]/pedigree/variants

Property: variants (Optional)

Type: Array

Description: Variants describe relations where the relationship between the components is not known. For example, if Component A contains nearly identical code to Component B. They are both related, but it is unclear if one is derived from the other, or if they share a common ancestor. Each item of this array must be a Component object.

8.1.27.3.1 Component

Location: /components/[]/pedigree/variants/[]

Type: Object

Reference: Refer to the component definition at /components/[]

8.1.27.4 Commits

Location: /components/[]/pedigree/commits

Property: commits (Optional)

Type: Array

Description: A list of zero or more commits which provide a trail describing how the component deviates from an ancestor, descendant, or variant. Each item of this array must be a Commit object.

8.1.27.4.1 Commit

Location: /components/[]/pedigree/commits/[]

Type: Object

Description: Specifies an individual commit

Table 58 – Properties for the commits object

Property	Type	Requirement	Description
uid	String	Optional	A unique identifier of the commit. This may be version control specific. For example, Subversion uses revision numbers whereas git uses commit hashes.
url	String	Optional	The URL to the commit. This URL will typically point to a commit in a version control system.
author	Object	Optional	The author who created the changes in the commit
committer	Object	Optional	The person who committed or pushed the commit
message	String	Optional	The text description of the contents of the commit

8.1.27.4.2 UID

Location: /components/[]/pedigree/commits/[]/uid
Property: uid (Optional)

Type: String

Description: A unique identifier of the commit. This may be version control specific. For example, Subversion uses revision numbers whereas git uses commit hashes.

8.1.27.4.3 URL

Location: /components/[]/pedigree/commits/[]/url
Property: url (Optional)

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL to the commit. This URL will typically point to a commit in a version control system.

8.1.27.4.4 Author

Location: /components/[]/pedigree/commits/[]/author
Property: author (Optional)

Type: Object

Description: The author who created the changes in the commit

Table 59 – Properties for the author object

Property	Type	Requirement	Description
timestamp	String	Optional	The timestamp in which the action occurred
name	String	Optional	The name of the individual who performed the action
email	String	Optional	The email address of the individual who performed the action

8.1.27.4.5 Timestamp

Location: /components/[]/pedigree/commits/[]/author/timestamp
Property: timestamp (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The timestamp in which the action occurred

8.1.27.4.6 Name

Location: /components/[]/pedigree/commits/[]/author/name
Property: name (Optional)

Type: String

Description: The name of the individual who performed the action

8.1.27.4.7 E-mail

Location: /components/[]/pedigree/commits/[]/author/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the individual who performed the action

8.1.27.4.8 Committer

Location: /components/[]/pedigree/commits/[]/committer
Property: committer (Optional)

Type: Object

Description: The person who committed or pushed the commit

Table 60 – Properties for the committer object

Property	Type	Requirement	Description
timestamp	String	Optional	The timestamp in which the action occurred
name	String	Optional	The name of the individual who performed the action
email	String	Optional	The email address of the individual who performed the action

8.1.27.4.9 Timestamp

Location: /components/[]/pedigree/commits/[]/committer/timestamp
Property: timestamp (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The timestamp in which the action occurred

8.1.27.4.10 Name

Location: /components/[]/pedigree/commits/[]/committer/name
Property: name (Optional)

Type: String

Description: The name of the individual who performed the action

8.1.27.4.11 E-mail

Location: /components/[]/pedigree/commits/[]/committer/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the individual who performed the action

8.1.27.4.12 Message

Location: /components/[]/pedigree/commits/[]/message
Property: message (Optional)

Type: String

Description: The text description of the contents of the commit

8.1.27.5 Patches

Location: /components/[]/pedigree/patches
Property: patches (Optional)

Type: Array

Description: >A list of zero or more patches describing how the component deviates from an ancestor, descendant, or variant. Patches may be complementary to commits or may be used in place of commits. Each item of this array must be a Patch object.

8.1.27.5.1 Patch

Location: /components/[]/pedigree/patches/[]

Type: Object
Description: Specifies an individual patch

Table 61 – Properties for the patches object

Property	Type	Requirement	Description
type	String	Required	Specifies the purpose for the patch including the resolution of defects, security issues, or new behavior or functionality.
diff	Object	Optional	The patch file (or diff) that shows changes. Refer to https://en.wikipedia.org/wiki/Diff
resolves	Array	Optional	A collection of issues the patch resolves

8.1.27.5.2 Patch Type

Location: /components/[]/pedigree/patches/[]/type
Property: type (Required)

Type: String

Description: Specifies the purpose for the patch including the resolution of defects, security issues, or new behaviour or functionality.

Table 62 – Enumeration of possible values

Value	Description
unofficial	A patch which is not developed by the creators or maintainers of the software being patched. Refer to https://en.wikipedia.org/wiki/Unofficial_patch.
monkey	A patch which dynamically modifies runtime behaviour. Refer to https://en.wikipedia.org/wiki/Monkey_patch.
backport	A patch which takes code from a newer version of the software and applies it to older versions of the same software. Refer to https://en.wikipedia.org/wiki/Backporting.
cherry-pick	A patch created by selectively applying commits from other versions or branches of the same software.

8.1.27.5.3 Diff

Location: /components/[]/pedigree/patches/[]/diff
Property: diff (Optional)

Type: Object

Description: The patch file (or diff) that shows changes. Refer to https://en.wikipedia.org/wiki/Diff

Table 63 – Properties for the diff object

Property	Type	Requirement	Description
text	Object	Optional	Specifies the optional text of the diff
url	String	Optional	Specifies the URL to the diff

8.1.27.5.4 Diff text

Location: /components/[]/pedigree/patches/[]/diff/text
Property: text (Optional)

Type: Object

Description: Specifies the optional text of the diff

Table 64 – Properties for the text object

Property	Type	Requirement	Description
contentType	String	Optional	Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.
encoding	String	Optional	Specifies the optional encoding the text is represented in.
content	String	Required	The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

8.1.27.5.5 Content-Type

Location: /components/[]/pedigree/patches/[]/diff/text/contentType
Property: contentType (Optional)

Type: String

Default Value: text/plain

Description: Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.

Examples:

- text/plain
- application/json

- image/png

8.1.27.5.6 Encoding

Location: /components/[]/pedigree/patches/[]/diff/text/encoding
Property: encoding (Optional)

Type: String

Description: Specifies the optional encoding the text is represented in.

Table 65 – Enumeration of possible values

Value	Description
base64	Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string.

8.1.27.5.7 Attachment Text

Location: /components/[]/pedigree/patches/[]/diff/text/content
Property: content (Required)

Type: String

Description: The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

8.1.27.5.8 URL

Location: /components/[]/pedigree/patches/[]/diff/url
Property: url (Optional)

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: Specifies the URL to the diff

8.1.27.5.9 Resolves

Location: /components/[]/pedigree/patches/[]/resolves
Property: resolves (Optional)

Type: Array

Description: A collection of issues the patch resolves Each item of this array must be an Issue object.

8.1.27.5.10 Issue

Location: /components/[]/pedigree/patches/[]/resolves/[]

Type: Object
Description: An individual issue that has been resolved.

Table 66 – Properties for the resolves object

Property	Type	Requirement	Description
type	String	Required	Specifies the type of issue
id	String	Optional	The identifier of the issue assigned by the source of the issue
name	String	Optional	The name of the issue
description	String	Optional	A description of the issue
source	Object	Optional	The source of the issue where it is documented
references	Array	Optional	A collection of URL's for reference. Multiple URLs are allowed.

8.1.27.5.11 Issue Type

Location: /components/[]/pedigree/patches/[]/resolves/[]/type
Property: type (Required)

Type: String

Description: Specifies the type of issue

Table 67 – Enumeration of possible values

Value	Description
defect	A fault, flaw, or bug in software.
enhancement	A new feature or behaviour in software.
security	A special type of defect which impacts security.

8.1.27.5.12 Issue ID

Location: /components/[]/pedigree/patches/[]/resolves/[]/id
Property: id (Optional)

Type: String

Description: The identifier of the issue assigned by the source of the issue

8.1.27.5.13 Issue Name

Location: /components/[]/pedigree/patches/[]/resolves/[]/name
Property: name (Optional)

Type: String

Description: The name of the issue

8.1.27.5.14 Issue Description

Location: /components/[]/pedigree/patches/[]/resolves/[]/description
Property: description (Optional)

Type: String

Description: A description of the issue

8.1.27.5.15 Source

Location: /components/[]/pedigree/patches/[]/resolves/[]/source
Property: source (Optional)

Type: Object

Description: The source of the issue where it is documented

Table 68 – Properties for the source object

Property	Type	Requirement	Description
name	String	Optional	The name of the source.
url	String	Optional	The url of the issue documentation as provided by the source

8.1.27.5.16 Name

Location: /components/[]/pedigree/patches/[]/resolves/[]/source/name
Property: name (Optional)

Type: String

Description: The name of the source.

Examples:

- National Vulnerability Database
- NVD
- Apache

8.1.27.5.17 URL

Location: /components/[]/pedigree/patches/[]/resolves/[]/source/url
Property: url (Optional)

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The url of the issue documentation as provided by the source

8.1.27.5.18 References

Location: /components/[]/pedigree/patches/[]/resolves/[]/references
Property: references (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: A collection of URL's for reference. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

8.1.27.6 Notes

Location: /components/[]/pedigree/notes
Property: notes (Optional)

Type: String

Description: Notes, observations, and other non-structured commentary describing the components pedigree.

8.1.28 External References

Location: /components/[]/externalReferences
Property: externalReferences (Optional)

Type: Array

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM. Each item of this array must be an External Reference object.

Reference: Refer to the external reference definition at /externalReferences/[]

8.1.28.1 External Reference

Location: /components/[]/externalReferences/[]

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

8.1.29 Components

Location: /components/[]/components
Property: components (Optional)

Type: Array

Description: A list of software and hardware components included in the parent component. This is not a dependency tree. It provides a way to specify a hierarchical representation of component assemblies, similar to system → subsystem → parts assembly in physical supply chains. Each item of this array must be a Component object.

8.1.29.1 Component

Location: /components/[]/components/[]

Type: Object
Reference: Refer to the component definition at /components/[]

All items must be unique.

8.1.30 Evidence

Location: /components/[]/evidence
Property: evidence (Optional)

Type: Object

Description: Provides the ability to document evidence collected through various forms of extraction or analysis.

Table 69 – Properties for the evidence object

Property	Type	Requirement	Description
identity	Array	Optional	Evidence that substantiates the identity of a component. The identity may be an object or an array of identity objects. Support for specifying identity as a single object was introduced in CycloneDX v1.5. Arrays were introduced in v1.6. It is recommended that all implementations use arrays, even if only one identity object is specified.
occurrences	Array	Optional	Evidence of individual instances of a component spread across multiple locations.
callstack	Object	Optional	Evidence of the components use through the callstack.
licenses	Array	Optional	EITHER (list of SPDX licenses and/or named licenses) OR (tuple of one SPDX License Expression)
copyright	Array	Optional	Copyright evidence captures intellectual property assertions, providing evidence of possible ownership and legal protection.

8.1.30.1 Identity Evidence

Location: /components/[]/evidence/identity
Property: identity (Optional)

Description: Evidence that substantiates the identity of a component. The identity may be an object or an array of identity objects. Support for specifying identity as a single object was introduced in CycloneDX v1.5. Arrays were introduced in v1.6. It is recommended that all implementations use arrays, even if only one identity object is specified.

*Must be **one of**:*

1. Array of Identity Objects
2. Identity Evidence

8.1.30.2 Array of Identity Objects

Type: Array

8.1.30.3 Identity Evidence

Type: Object
Description: Evidence that substantiates the identity of a component.

Table 70 – Properties for identity evidence

Property	Type	Requirement	Description
field	String	Required	The identity field of the component which the evidence describes.
confidence	Number	Optional	The overall confidence of the evidence from 0 - 1, where 1 is 100% confidence.
concludedValue	String	Optional	The value of the field (cpe, purl, etc) that has been concluded based on the aggregate of all methods (if available).
methods	Array	Optional	The methods used to extract and/or analyze the evidence.

Property	Type	Requirement	Description
tools	Array	Optional	The object in the BOM identified by its bom-ref. This is often a component or service but may be any object type supporting bom-refs. Tools used for analysis should already be defined in the BOM, either in the metadata/tools, components, or formulation.

8.1.30.4 Array of Identity Objects

Location: /components/[]/evidence/identity
Property: identity

Type: Array

8.1.30.4.1 Identity Evidence

Location: /components/[]/evidence/identity/[]

Type: Object
Description: Evidence that substantiates the identity of a component.

Table 71 – Properties for the identity object

Property	Type	Requirement	Description
field	String	Required	The identity field of the component which the evidence describes.
confidence	Number	Optional	The overall confidence of the evidence from 0 - 1, where 1 is 100% confidence.
concludedValue	String	Optional	The value of the field (cpe, purl, etc) that has been concluded based on the aggregate of all methods (if available).
methods	Array	Optional	The methods used to extract and/or analyze the evidence.
tools	Array	Optional	The object in the BOM identified by its bom-ref. This is often a component or service but may be any object type supporting bom-refs. Tools used for analysis should already be defined in the BOM, either in the metadata/tools, components, or formulation.

8.1.30.4.2 Field

Location: /components/[]/evidence/identity/[]/field
Property: field (Required)

Type: String

Description: The identity field of the component which the evidence describes.

Enumeration: Must be one of:

- group
- name
- version
- purl
- cpe

- omniborId
- swhid
- swid
- hash

8.1.30.4.3 Confidence

Location: /components/[]/evidence/identity/[]/confidence
Property: confidence (Optional)

Type: Number

Maximum Value: 1

Description: The overall confidence of the evidence from 0 - 1, where 1 is 100% confidence.

8.1.30.4.4 Concluded Value

Location: /components/[]/evidence/identity/[]/concludedValue
Property: concludedValue (Optional)

Type: String

Description: The value of the field (cpe, purl, etc) that has been concluded based on the aggregate of all methods (if available).

8.1.30.4.5 Methods

Location: /components/[]/evidence/identity/[]/methods
Property: methods (Optional)

Type: Array

Description: The methods used to extract and/or analyze the evidence.

Location: /components/[]/evidence/identity/[]/methods/[]

Type: Object

Table 72 – Properties for the methods object

Property	Type	Requirement	Description
technique	String	Required	The technique used in this method of analysis.
confidence	Number	Required	The confidence of the evidence from 0 - 1, where 1 is 100% confidence. Confidence is specific to the technique used. Each technique of analysis can have independent confidence.
value	String	Optional	The value or contents of the evidence.

8.1.30.4.6 Technique

Location: /components/[]/evidence/identity/[]/methods/[]/technique
Property: technique (Required)

Type: String

Description: The technique used in this method of analysis.

Enumeration: Must be one of:

- source-code-analysis

- binary-analysis
- manifest-analysis
- ast-fingerprint
- hash-comparison
- instrumentation
- dynamic-analysis
- filename
- attestation
- other

8.1.30.4.7 Confidence

Location: /components/[]/evidence/identity/[]/methods/[]/confidence
Property: confidence (Required)

Type: Number

Maximum Value: 1

Description: The confidence of the evidence from 0 - 1, where 1 is 100% confidence. Confidence is specific to the technique used. Each technique of analysis can have independent confidence.

8.1.30.4.8 Value

Location: /components/[]/evidence/identity/[]/methods/[]/value
Property: value (Optional)

Type: String

Description: The value or contents of the evidence.

8.1.30.4.9 BOM References

Location: /components/[]/evidence/identity/[]/tools
Property: tools (Optional)

Type: Array

Description: The object in the BOM identified by its bom-ref. This is often a component or service but may be any object type supporting bom-refs. Tools used for analysis should already be defined in the BOM, either in the metadata/tools, components, or formulation.

Location: /components/[]/evidence/identity/[]/tools/[]

*Must be **any of**:*

1. Ref
2. BOM-Link Element

8.1.30.4.10 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

8.1.30.4.11 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

All items must be unique.

8.1.30.4.12 Field

Location: /components/[]/evidence/identity/field
Property: identity (Required)

Type: String

Description: The identity field of the component which the evidence describes.

Enumeration: Must be one of:

- group
- name
- version
- purl
- cpe
- omniborId
- swhid
- swid
- hash

8.1.30.4.13 Confidence

Location: /components/[]/evidence/identity/confidence
Property: identity (Optional)

Type: Number

Maximum Value: 1

Description: The overall confidence of the evidence from 0 - 1, where 1 is 100% confidence.

8.1.30.4.14 Concluded Value

Location: /components/[]/evidence/identity/concludedValue
Property: identity (Optional)

Type: String

Description: The value of the field (cpe, purl, etc) that has been concluded based on the aggregate of all methods (if available).

8.1.30.4.15 Methods

Location: /components/[]/evidence/identity/methods
Property: identity (Optional)

Type: Array

Description: The methods used to extract and/or analyze the evidence.

Location: /components/[]/evidence/identity/methods/[]

Type: Object

Table 73 – Properties for the identity object

Property	Type	Requirement	Description
technique	String	Required	The technique used in this method of analysis.
confidence	Number	Required	The confidence of the evidence from 0 - 1, where 1 is 100% confidence. Confidence is specific to the technique used. Each technique of analysis can have independent confidence.
value	String	Optional	The value or contents of the evidence.

8.1.30.4.16 Technique

Location: /components/[]/evidence/identity/methods/[]/technique
Property: technique (Required)

Type: String

Description: The technique used in this method of analysis.

Enumeration: Must be one of:

- source-code-analysis
- binary-analysis
- manifest-analysis
- ast-fingerprint
- hash-comparison
- instrumentation
- dynamic-analysis
- filename
- attestation
- other

8.1.30.4.17 Confidence

Location: /components/[]/evidence/identity/methods/[]/confidence
Property: confidence (Required)

Type: Number

Maximum Value: 1

Description: The confidence of the evidence from 0 - 1, where 1 is 100% confidence. Confidence is specific to the technique used. Each technique of analysis can have independent confidence.

8.1.30.4.18 Value

Location: /components/[]/evidence/identity/methods/[]/value
Property: value (Optional)

Type: String

Description: The value or contents of the evidence.

8.1.30.4.19 BOM References

Location: /components/[]/evidence/identity/tools
Property: identity (Optional)

Type: Array

Description: The object in the BOM identified by its bom-ref. This is often a component or service but may be any object type supporting bom-refs. Tools used for analysis should already be defined in the BOM, either in the metadata/tools, components, or formulation.

Location: /components/[]/evidence/identity/tools/[]

*Must be **any of**:*

1. Ref
2. BOM-Link Element

8.1.30.4.20 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

8.1.30.4.21 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

All items must be unique.

8.1.30.5 Occurrences

Location: /components/[]/evidence/occurrences
Property: occurrences (Optional)

Type: Array

Description: Evidence of individual instances of a component spread across multiple locations.

Location: /components/[]/evidence/occurrences/[]

Type: Object

Table 74 – Properties for the occurrences object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the occurrence elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
location	String	Required	The location or path to where the component was found.
line	Integer	Optional	The line number where the component was found.
offset	Integer	Optional	The offset where the component was found.
symbol	String	Optional	The symbol name that was found associated with the component.

Property	Type	Requirement	Description
additionalContext	String	Optional	Any additional context of the detected component (e.g. a code snippet).

8.1.30.5.1 BOM Reference

Location: /components/[]/evidence/occurrences/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the occurrence elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.30.5.2 Location

Location: /components/[]/evidence/occurrences/[]/location
Property: location (Required)

Type: String

Description: The location or path to where the component was found.

8.1.30.5.3 Line Number

Location: /components/[]/evidence/occurrences/[]/line
Property: line (Optional)

Type: Integer

Description: The line number where the component was found.

8.1.30.5.4 Offset

Location: /components/[]/evidence/occurrences/[]/offset
Property: offset (Optional)

Type: Integer

Description: The offset where the component was found.

8.1.30.5.5 Symbol

Location: /components/[]/evidence/occurrences/[]/symbol
Property: symbol (Optional)

Type: String

Description: The symbol name that was found associated with the component.

8.1.30.5.6 Additional Context

Location: /components/[]/evidence/occurrences/[]/additionalContext
Property: additionalContext (Optional)

Type: String

Description: Any additional context of the detected component (e.g. a code snippet).

8.1.30.6 Call Stack

Location: /components/[]/evidence/callstack
Property: callstack (Optional)

Type: Object

Description: Evidence of the components use through the callstack.

Table 75 – Properties for the callstack object

Property	Type	Requirement	Description
frames	Array	Optional	Within a call stack, a frame is a discrete unit that encapsulates an execution context, including local variables, parameters, and the return address. As function calls are made, frames are pushed onto the stack, forming an array-like structure that orchestrates the flow of program execution and manages the sequence of function invocations.

8.1.30.6.1 Frames

Location: /components/[]/evidence/callstack/frames
Property: frames (Optional)

Type: Array

Description: Within a call stack, a frame is a discrete unit that encapsulates an execution context, including local variables, parameters, and the return address. As function calls are made, frames are pushed onto the stack, forming an array-like structure that orchestrates the flow of programme execution and manages the sequence of function invocations.

Location: /components/[]/evidence/callstack/frames/[]

Type: Object

Table 76 – Properties for the frames object

Property	Type	Requirement	Description
package	String	Optional	A package organizes modules into namespaces, providing a unique namespace for each type it contains.
module	String	Required	A module or class that encloses functions/methods and other code.
function	String	Optional	A block of code designed to perform a particular task.
parameters	Array	Optional	Optional arguments that are passed to the module or function.
line	Integer	Optional	The line number the code that is called resides on.
column	Integer	Optional	The column the code that is called resides.
fullFilename	String	Optional	The full path and filename of the module.

8.1.30.6.2 Package

Location: /components/[]/evidence/callstack/frames/[]/package
Property: package (Optional)

Type: String

Description: A package organizes modules into namespaces, providing a unique namespace for each type it contains.

8.1.30.6.3 Module

Location: /components/[]/evidence/callstack/frames/[]/module
Property: module (Required)

Type: String

Description: A module or class that encloses functions/methods and other code.

8.1.30.6.4 Function

Location: /components/[]/evidence/callstack/frames/[]/function
Property: function (Optional)

Type: String

Description: A block of code designed to perform a particular task.

8.1.30.6.5 Parameters

Location: /components/[]/evidence/callstack/frames/[]/parameters
Property: parameters (Optional)

Type: array (of String)

Description: Optional arguments that are passed to the module or function. Each item of this array must be a string.

8.1.30.6.6 Line

Location: /components/[]/evidence/callstack/frames/[]/line
Property: line (Optional)

Type: Integer

Description: The line number the code that is called resides on.

8.1.30.6.7 Column

Location: /components/[]/evidence/callstack/frames/[]/column
Property: column (Optional)

Type: Integer

Description: The column the code that is called resides.

8.1.30.6.8 Full Filename

Location: /components/[]/evidence/callstack/frames/[]/fullFilename
Property: fullFilename (Optional)

Type: String

Description: The full path and filename of the module.

8.1.30.7 Licence Evidence

Location: /components/[]/evidence/licenses
Property: licenses (Optional)

Type: Array

Description: EITHER (list of SPDX licences and/or named licences) OR (tuple of one SPDX Licence Expression)

Reference: Refer to the license definition at /components/[]/licenses

8.1.30.8 Copyright Evidence

Location: /components/[]/evidence/copyright
Property: copyright (Optional)

Type: Array

Description: Copyright evidence captures intellectual property assertions, providing evidence of possible ownership and legal protection. Each item of this array must be a Copyright object.

8.1.30.8.1 Copyright

Location: /components/[]/evidence/copyright/[]

Type: Object

Description: A copyright notice informing users of the underlying claims to copyright ownership in a published work.

Table 77 – Properties for the copyright object

Property	Type	Requirement	Description
text	String	Required	The textual content of the copyright.

8.1.30.8.2 Copyright Text

Location: /components/[]/evidence/copyright/[]/text
Property: text (Required)

Type: String

Description: The textual content of the copyright.

8.1.31 Release notes

Location: /components/[]/releaseNotes
Property: releaseNotes (Optional)

Type: Object

Description: Specifies optional release notes.

Table 78 – Properties for the releaseNotes object

Property	Type	Requirement	Description
type	String	Required	The software versioning type the release note describes.
title	String	Optional	The title of the release.
featuredImage	String	Optional	The URL to an image that may be prominently displayed with the release note.
socialImage	String	Optional	The URL to an image that may be used in messaging on social media platforms.
description	String	Optional	A short description of the release.
timestamp	String	Optional	The date and time (timestamp) when the release note was created.

Property	Type	Requirement	Description
aliases	Array	Optional	One or more alternate names the release may be referred to. This may include unofficial terms used by development and marketing teams (e.g. code names).
tags	Array	Optional	Textual strings that aid in discovery, search, and retrieval of the associated object. Tags often serve as a way to group or categorize similar or related objects by various attributes.
resolves	Array	Optional	A collection of issues that have been resolved.
notes	Array	Optional	Zero or more release notes containing the locale and content. Multiple note objects may be specified to support release notes in a wide variety of languages.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

8.1.31.1 Type

Location: /components/[]/releaseNotes/type
Property: type (Required)

Type: String

Description: The software versioning type the release note describes.

Examples:

- major
- minor
- patch
- pre-release
- internal

8.1.31.2 Title

Location: /components/[]/releaseNotes/title
Property: title (Optional)

Type: String

Description: The title of the release.

8.1.31.3 Featured image

Location: /components/[]/releaseNotes/featuredImage
Property: featuredImage (Optional)

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL to an image that may be prominently displayed with the release note.

8.1.31.4 Social image

Location: /components/[]/releaseNotes/socialImage
Property: socialImage (Optional)

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL to an image that may be used in messaging on social media platforms.

8.1.31.5 Description

Location: /components/[]/releaseNotes/description
Property: description (Optional)

Type: String

Description: A short description of the release.

8.1.31.6 Timestamp

Location: /components/[]/releaseNotes/timestamp
Property: timestamp (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time (timestamp) when the release note was created.

8.1.31.7 Aliases

Location: /components/[]/releaseNotes/aliases
Property: aliases (Optional)

Type: array (of String)

Description: One or more alternate names the release may be referred to. This may include unofficial terms used by development and marketing teams (e.g. code names). Each item of this array must be a string.

8.1.31.8 Tags

Location: /components/[]/releaseNotes/tags
Property: tags (Optional)

Type: array (of String)

Description: Textual strings that aid in discovery, search, and retrieval of the associated object. Tags often serve as a way to group or categorize similar or related objects by various attributes. Each item of this array must be a string.

Examples:

- json-parser
- object-persistence
- text-to-image
- translation
- object-detection

8.1.31.9 Resolves

Location: /components/[]/releaseNotes/resolves

Property: resolves (Optional)

Type: Array

Description: A collection of issues that have been resolved. Each item of this array must be an Issue object.

8.1.31.9.1 Issue

Location: /components/[]/releaseNotes/resolves/[]

Type: Object

Description: An individual issue that has been resolved.

Table 79 – Properties for the resolves object

Property	Type	Requirement	Description
type	String	Required	Specifies the type of issue
id	String	Optional	The identifier of the issue assigned by the source of the issue
name	String	Optional	The name of the issue
description	String	Optional	A description of the issue
source	Object	Optional	The source of the issue where it is documented
references	Array	Optional	A collection of URL's for reference. Multiple URLs are allowed.

8.1.31.9.2 Issue Type

Location: /components/[]/releaseNotes/resolves/[]/type
Property: type (Required)

Type: String

Description: Specifies the type of issue

Table 80 – Enumeration of possible values

Value	Description
defect	A fault, flaw, or bug in software.
enhancement	A new feature or behaviour in software.
security	A special type of defect which impacts security.

8.1.31.9.3 Issue ID

Location: /components/[]/releaseNotes/resolves/[]/id
Property: id (Optional)

Type: String

Description: The identifier of the issue assigned by the source of the issue

8.1.31.9.4 Issue Name

Location: /components/[]/releaseNotes/resolves/[]/name
Property: name (Optional)

Type: String

Description: The name of the issue

8.1.31.9.5 Issue Description

Location: /components/[]/releaseNotes/resolves/[]/description
Property: description (Optional)

Type: String

Description: A description of the issue

8.1.31.9.6 Source

Location: /components/[]/releaseNotes/resolves/[]/source
Property: source (Optional)

Type: Object

Description: The source of the issue where it is documented

Table 81 – Properties for the source object

Property	Type	Requirement	Description
name	String	Optional	The name of the source.
url	String	Optional	The url of the issue documentation as provided by the source

8.1.31.9.7 Name

Location: /components/[]/releaseNotes/resolves/[]/source/name
Property: name (Optional)

Type: String

Description: The name of the source.

Examples:

- National Vulnerability Database
- NVD
- Apache

8.1.31.9.8 URL

Location: /components/[]/releaseNotes/resolves/[]/source/url
Property: url (Optional)

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The url of the issue documentation as provided by the source

8.1.31.9.9 References

Location: /components/[]/releaseNotes/resolves/[]/references

Property: references (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: A collection of URL's for reference. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

8.1.31.10 Notes

Location: /components/[]/releaseNotes/notes
Property: notes (Optional)

Type: Array

Description: Zero or more release notes containing the locale and content. Multiple note objects may be specified to support release notes in a wide variety of languages. Each item of this array must be a Note object.

8.1.31.10.1 Note

Location: /components/[]/releaseNotes/notes/[]

Type: Object
Description: A note containing the locale and content.

Table 82 – Properties for the notes object

Property	Type	Requirement	Description
locale	String	Optional	The ISO-639 (or higher) language code and optional ISO-3166 (or higher) country code. Examples include: "en", "en-US", "fr" and "fr-CA"
text	Object	Required	Specifies the full content of the release note.

8.1.31.10.2 Locale

Location: /components/[]/releaseNotes/notes/[]/locale
Property: locale (Optional)

Type: String

Pattern Constraint: ^([a-z]{2})(-[A-Z]{2})?$

Description: The ISO-639 (or higher) language code and optional ISO-3166 (or higher) country code. Examples include: "en", "en-US", "fr" and "fr-CA"

8.1.31.10.3 Release note content

Location: /components/[]/releaseNotes/notes/[]/text
Property: text (Required)

Type: Object

Description: Specifies the full content of the release note.

Table 83 – Properties for the text object

Property	Type	Requirement	Description
contentType	String	Optional	Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.
encoding	String	Optional	Specifies the optional encoding the text is represented in.
content	String	Required	The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

8.1.31.10.4 Content-Type

Location: /components/[]/releaseNotes/notes/[]/text/contentType
Property: contentType (Optional)

Type: String

Default Value: text/plain

Description: Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.

Examples:

- text/plain
- application/json
- image/png

8.1.31.10.5 Encoding

Location: /components/[]/releaseNotes/notes/[]/text/encoding
Property: encoding (Optional)

Type: String

Description: Specifies the optional encoding the text is represented in.

Table 84 – Enumeration of possible values

Value	Description
base64	Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string.

8.1.31.10.6 Attachment Text

Location: /components/[]/releaseNotes/notes/[]/text/content
Property: content (Required)

Type: String

Description: The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

8.1.31.11 Properties

Location: /components/[]/releaseNotes/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

8.1.31.11.1 Lightweight name-value pair

Location: /components/[]/releaseNotes/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 85 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

8.1.31.11.2 Name

Location: /components/[]/releaseNotes/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

8.1.31.11.3 Value

Location: /components/[]/releaseNotes/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

8.1.32 AI/ML Model Card

Location: /components/[]/modelCard
Property: modelCard (Optional)

Type: Object

Description: A model card describes the intended uses of a machine learning model and potential limitations, including biases and ethical considerations. Model cards typically contain the training parameters, which datasets were used to train the model, performance metrics, and other relevant data useful for ML transparency. This object SHOULD be specified for any component of type machine-learning-model and must not be specified for other component types.

Table 86 – Properties for the modelCard object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the model card elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
modelParameters	Object	Optional	Hyper-parameters for construction of the model.
quantitativeAnalysis	Object	Optional	A quantitative analysis of the model
considerations	Object	Optional	What considerations should be taken into account regarding the model's construction, training, and application?
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

8.1.32.1 BOM Reference

Location: /components/[]/modelCard/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the model card elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.32.2 Model Parameters

Location: /components/[]/modelCard/modelParameters
Property: modelParameters (Optional)

Type: Object

Description: Hyper-parameters for construction of the model.

Table 87 – Properties for the modelParameters object

Property	Type	Requirement	Description
approach	Object	Optional	The overall approach to learning used by the model for problem solving.
task	String	Optional	Directly influences the input and/or output. Examples include classification, regression, clustering, etc.
architectureFamily	String	Optional	The model architecture family such as transformer network, convolutional neural network, residual neural network, LSTM neural network, etc.
modelArchitecture	String	Optional	The specific architecture of the model such as GPT-1, ResNet-50, YOLOv3, etc.
datasets	Array	Optional	The datasets used to train and evaluate the model.
inputs	Array	Optional	The input format(s) of the model
outputs	Array	Optional	The output format(s) from the model

8.1.32.2.1 Approach

Location: /components/[]/modelCard/modelParameters/approach
Property: approach (Optional)

Type: Object

Description: The overall approach to learning used by the model for problem solving.

Table 88 – Properties for the approach object

Property	Type	Requirement	Description
type	String	Optional	Learning types describing the learning problem or hybrid learning problem.

8.1.32.2.2 Learning Type

Location: /components/[]/modelCard/modelParameters/approach/type
Property: type (Optional)

Type: String

Description: Learning types describing the learning problem or hybrid learning problem.

Table 89 – Enumeration of possible values

Value	Description
supervised	Supervised machine learning involves training an algorithm on labelled data to predict or classify new data based on the patterns learned from the labelled examples.
unsupervised	Unsupervised machine learning involves training algorithms on unlabeled data to discover patterns, structures, or relationships without explicit guidance, allowing the model to identify inherent structures or clusters within the data.
reinforcement-learning	Reinforcement learning is a type of machine learning where an agent learns to make decisions by interacting with an environment to maximize cumulative rewards, through trial and error.
semi-supervised	Semi-supervised machine learning utilizes a combination of labelled and unlabeled data during training to improve model performance, leveraging the benefits of both supervised and unsupervised learning techniques.
self-supervised	Self-supervised machine learning involves training models to predict parts of the input data from other parts of the same data, without requiring external labels, enabling learning from large amounts of unlabeled data.

8.1.32.2.3 Task

Location: /components/[]/modelCard/modelParameters/task
Property: task (Optional)

Type: String

Description: Directly influences the input and/or output. Examples include classification, regression, clustering, etc.

8.1.32.2.4 Architecture Family

Location: /components/[]/modelCard/modelParameters/architectureFamily
Property: architectureFamily (Optional)

Type: String

Description: The model architecture family such as transformer network, convolutional neural network, residual neural network, LSTM neural network, etc.

8.1.32.2.5 Model Architecture

Location: /components/[]/modelCard/modelParameters/modelArchitecture
Property: modelArchitecture (Optional)

Type: String

Description: The specific architecture of the model such as GPT-1, ResNet-50, YOLOv3, etc.

8.1.32.2.6 Datasets

Location: /components/[]/modelCard/modelParameters/datasets
Property: datasets (Optional)

Type: Array

Description: The datasets used to train and evaluate the model.

Location: /components/[]/modelCard/modelParameters/datasets/[]

*Must be **one of**:*

1. Inline Data Information
2. Data Reference

8.1.32.2.7 Inline Data Information

Type: Object

Table 90 – Properties for inline data information

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the dataset elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
type	String	Required	The general theme or subject matter of the data being specified.
name	String	Optional	The name of the dataset.
contents	Object	Optional	The contents or references to the contents of the data being described.
classification	String	Optional	Data classification tags data according to its type, sensitivity, and value if altered, stolen, or destroyed.
sensitiveData	Array	Optional	A description of any sensitive data in a dataset.
graphics	Object	Optional	A collection of graphics that represent various measurements.
description	String	Optional	A description of the dataset. Can describe size of dataset, whether it's used for source code, training, testing, or validation, etc.
governance	Object	Optional	Data governance captures information regarding data ownership, stewardship, and custodianship, providing insights into the individuals or entities responsible for managing, overseeing, and safeguarding the data throughout its lifecycle.

8.1.32.2.8 Data Reference

Type: Object

Table 91 – Properties for data reference

Property	Type	Requirement	Description
ref	String	Optional	References a data component by the components bom-ref attribute

8.1.32.2.9 BOM Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/bom-ref
Property: datasets (Optional)

Type: String

Description: An optional identifier which can be used to reference the dataset elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.32.2.10 Type of Data

Location: /components/[]/modelCard/modelParameters/datasets/[]/type
Property: datasets (Required)

Type: String

Description: The general theme or subject matter of the data being specified.

Table 92 – Enumeration of possible values

Value	Description
source-code	Any type of code, code snippet, or data-as-code.
configuration	Parameters or settings that may be used by other components.
dataset	A collection of data.
definition	Data that can be used to create new instances of what the definition defines.
other	Any other type of data that does not fit into existing definitions.

8.1.32.2.11 Dataset Name

Location: /components/[]/modelCard/modelParameters/datasets/[]/name
Property: datasets (Optional)

Type: String

Description: The name of the dataset.

8.1.32.2.12 Data Contents

Location: /components/[]/modelCard/modelParameters/datasets/[]/contents
Property: datasets (Optional)

Type: Object

Description: The contents or references to the contents of the data being described.

Table 93 – Properties for the datasets object

Property	Type	Requirement	Description
attachment	Object	Optional	An optional way to include textual or encoded data.
url	String	Optional	The URL to where the data can be retrieved.
properties	Array	Optional	Provides the ability to document name-value parameters used for configuration.

8.1.32.2.13 Data Attachment

Location: /components/[]/modelCard/modelParameters/datasets/[]/contents/attachment
Property: attachment (Optional)

Type: Object

Description: An optional way to include textual or encoded data.

Table 94 – Properties for the attachment object

Property	Type	Requirement	Description
contentType	String	Optional	Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.
encoding	String	Optional	Specifies the optional encoding the text is represented in.
content	String	Required	The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

8.1.32.2.14 Content-Type

Location: /components/[]/modelCard/modelParameters/datasets/[]/contents/attachment/contentType
Property: contentType (Optional)

Type: String

Default Value: text/plain

Description: Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.

Examples:

- text/plain
- application/json
- image/png

8.1.32.2.15 Encoding

Location: /components/[]/modelCard/modelParameters/datasets/[]/contents/attachment/encoding
Property: encoding (Optional)

Type: String

Description: Specifies the optional encoding the text is represented in.

Table 95 – Enumeration of possible values

Value	Description
base64	Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string.

8.1.32.2.16 Attachment Text

Location: /components/[]/modelCard/modelParameters/datasets/[]/contents/attachment/content
Property: content (Required)

Type: String

Description: The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

8.1.32.2.17 Data URL

Location: /components/[]/modelCard/modelParameters/datasets/[]/contents/url
Property: url (Optional)

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL to where the data can be retrieved.

8.1.32.2.18 Configuration Properties

Location: /components/[]/modelCard/modelParameters/datasets/[]/contents/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document name-value parameters used for configuration. Each item of this array must be a Lightweight name-value pair object.

8.1.32.2.19 Lightweight name-value pair

Location: /components/[]/modelCard/modelParameters/datasets/[]/contents/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 96 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

8.1.32.2.20 Name

Location: /components/[]/modelCard/modelParameters/datasets/[]/contents/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

8.1.32.2.21 Value

Location: /components/[]/modelCard/modelParameters/datasets/[]/contents/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

8.1.32.2.22 Data Classification

Location: /components/[]/modelCard/modelParameters/datasets/[]/classification
Property: datasets (Optional)

Type: String

Description: Data classification tags data according to its type, sensitivity, and value if altered, stolen, or destroyed.

8.1.32.2.23 Sensitive Data

Location: /components/[]/modelCard/modelParameters/datasets/[]/sensitiveData
Property: datasets (Optional)

Type: array (of String)

Description: A description of any sensitive data in a dataset. Each item of this array must be a string.

8.1.32.2.24 Graphics Collection

Location: /components/[]/modelCard/modelParameters/datasets/[]/graphics
Property: datasets (Optional)

Type: Object

Description: A collection of graphics that represent various measurements.

Table 97 – Properties for the datasets object

Property	Type	Requirement	Description
description	String	Optional	A description of this collection of graphics.
collection	Array	Optional	A collection of graphics.

8.1.32.2.25 Description

Location: /components/[]/modelCard/modelParameters/datasets/[]/graphics/description
Property: description (Optional)

Type: String

Description: A description of this collection of graphics.

8.1.32.2.26 Collection

Location: /components/[]/modelCard/modelParameters/datasets/[]/graphics/collection
Property: collection (Optional)

Type: Array

Description: A collection of graphics. Each item of this array must be a Graphic object.

8.1.32.2.27 Graphic

Location: /components/[]/modelCard/modelParameters/datasets/[]/graphics/collection/[]

Type: Object

Table 98 – Properties for the collection object

Property	Type	Requirement	Description
name	String	Optional	The name of the graphic.
image	Object	Optional	The graphic (vector or raster). Base64 encoding must be specified for binary images.

8.1.32.2.28 Name

Location: /components/[]/modelCard/modelParameters/datasets/[]/graphics/collection/[]/name
Property: name (Optional)

Type: String

Description: The name of the graphic.

8.1.32.2.29 Graphic Image

Location: /components/[]/modelCard/modelParameters/datasets/[]/graphics/collection/[]/image
Property: image (Optional)

Type: Object

Description: The graphic (vector or raster). Base64 encoding must be specified for binary images.

Table 99 – Properties for the image object

Property	Type	Requirement	Description
contentType	String	Optional	Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.
encoding	String	Optional	Specifies the optional encoding the text is represented in.
content	String	Required	The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

8.1.32.2.30 Content-Type

Location: /components/[]/modelCard/modelParameters/datasets/[]/graphics/collection/[]/image/contentType
Property: contentType (Optional)

Type: String

Default Value: text/plain

Description: Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.

Examples:

- text/plain
- application/json
- image/png

8.1.32.2.31 Encoding

Location: /components/[]/modelCard/modelParameters/datasets/[]/graphics/collection/[]/image/encoding
Property: encoding (Optional)

Type: String

Description: Specifies the optional encoding the text is represented in.

Table 100 – Enumeration of possible values

Value	Description
base64	Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string.

8.1.32.2.32 Attachment Text

Location: /components/[]/modelCard/modelParameters/datasets/[]/graphics/collection/[]/image/content
Property: content (Required)

Type: String

Description: The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

8.1.32.2.33 Dataset Description

Location: /components/[]/modelCard/modelParameters/datasets/[]/description
Property: datasets (Optional)

Type: String

Description: A description of the dataset. Can describe size of dataset, whether it's used for source code, training, testing, or validation, etc.

8.1.32.2.34 Data Governance

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance
Property: datasets (Optional)

Type: Object

Description: Data governance captures information regarding data ownership, stewardship, and custodianship, providing insights into the individuals or entities responsible for managing, overseeing, and safeguarding the data throughout its lifecycle.

Table 101 – Properties for the datasets object

Property	Type	Requirement	Description
custodians	Array	Optional	Data custodians are responsible for the safe custody, transport, and storage of data.
stewards	Array	Optional	Data stewards are responsible for data content, context, and associated business rules.
owners	Array	Optional	Data owners are concerned with risk and appropriate access to data.

8.1.32.2.35 Data Custodians

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians
Property: custodians (Optional)

Type: Array

Description: Data custodians are responsible for the safe custody, transport, and storage of data.

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]

Type: Object

Table 102 – Properties for the custodians object

Property	Type	Requirement	Description
organization	Object	Optional	The organization that is responsible for specific data governance role(s).
contact	Object	Optional	The individual that is responsible for specific data governance role(s).

8.1.32.2.36 Organization

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization
Property: organization (Optional)

Type: Object

Description: The organization that is responsible for specific data governance role(s).

Table 103 – Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

8.1.32.2.37 BOM Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/bom-ref

Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.32.2.38 Organization Name

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/name

Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

8.1.32.2.39 Organization Address

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/address

Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 104 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

8.1.32.2.40 BOM Reference

Location:
/components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/address/bom-ref

Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.32.2.41 Country

Location:
/components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/address/country

Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

8.1.32.2.42 Region

Location:
/components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/address/region

Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

8.1.32.2.43 Locality

Location:
/components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/address/locality

Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

8.1.32.2.44 Post Office Box Number

Location:
/components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/address/postOfficeBoxNumber

Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

8.1.32.2.45 Postal Code

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/address/postalCode

Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

8.1.32.2.46 Street Address

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/address/streetAddress

Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

8.1.32.2.47 Organization URL(s)

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/url

Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

8.1.32.2.48 Organizational Contact

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/contact

Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

8.1.32.2.49 Organizational Contact

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/contact/[]

Type: Object

Table 105 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

8.1.32.2.50 BOM Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/contact/[]/bom-ref

Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.32.2.51 Name

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/contact/[]/name

Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

8.1.32.2.52 Email Address

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/contact/[]/email

Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

8.1.32.2.53 Phone

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/contact/[]/phone

Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

8.1.32.2.54 Organizational Contact

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/contact

Property: contact (Optional)

Type: Object

Description: The individual that is responsible for specific data governance role(s).

Table 106 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

8.1.32.2.55 BOM Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/contact/bom-ref

Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.32.2.56 Name

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/contact/name

Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

8.1.32.2.57 Email Address

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/contact/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

8.1.32.2.58 Phone

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/contact/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

8.1.32.2.59 Data Stewards

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards
Property: stewards (Optional)

Type: Array

Description: Data stewards are responsible for data content, context, and associated business rules.

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]

Type: Object

Table 107 – Properties for the stewards object

Property	Type	Requirement	Description
organization	Object	Optional	The organization that is responsible for specific data governance role(s).
contact	Object	Optional	The individual that is responsible for specific data governance role(s).

8.1.32.2.60 Organization

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization
Property: organization (Optional)

Type: Object

Description: The organization that is responsible for specific data governance role(s).

Table 108 – Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

8.1.32.2.61 BOM Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/bom-ref

Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.32.2.62 Organization Name

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/name

Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

8.1.32.2.63 Organization Address

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/address

Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 109 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

8.1.32.2.64 BOM Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/address/bom-ref

Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.32.2.65 Country

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/address/country

Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

8.1.32.2.66 Region

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/address/region

Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

8.1.32.2.67 Locality

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/address/locality

Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

8.1.32.2.68 Post Office Box Number

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/address/postOfficeBoxNumber

Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

8.1.32.2.69 Postal Code

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/address/postalCode

Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

8.1.32.2.70 Street Address

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/address/streetAddress

Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

8.1.32.2.71 Organization URL(s)

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/url

Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

8.1.32.2.72 Organizational Contact

Location:
/components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/contact

Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

8.1.32.2.73 Organizational Contact

Location:
/components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/contact/[]

Type: Object

Table 110 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

8.1.32.2.74 BOM Reference

Location:
/components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/contact/[]/bom-ref

Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.32.2.75 Name

Location:
/components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/contact/[]/name

Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

8.1.32.2.76 Email Address

Location:
/components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/contact/[]/email

Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

8.1.32.2.77 Phone

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

8.1.32.2.78 Organizational Contact

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/contact
Property: contact (Optional)

Type: Object

Description: The individual that is responsible for specific data governance role(s).

Table 111 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

8.1.32.2.79 BOM Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/contact/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.32.2.80 Name

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/contact/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

8.1.32.2.81 Email Address

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/contact/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

8.1.32.2.82 Phone

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/contact/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

8.1.32.2.83 Data Owners

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners
Property: owners (Optional)

Type: Array

Description: Data owners are concerned with risk and appropriate access to data.

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]

Type: Object

Table 112 – Properties for the owners object

Property	Type	Requirement	Description
organization	Object	Optional	The organization that is responsible for specific data governance role(s).
contact	Object	Optional	The individual that is responsible for specific data governance role(s).

8.1.32.2.84 Organization

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization
Property: organization (Optional)

Type: Object

Description: The organization that is responsible for specific data governance role(s).

Table 113 – Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

8.1.32.2.85 BOM Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.32.2.86 Organization Name

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/name
Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

8.1.32.2.87 Organization Address

Location:
/components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/address
Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 114 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

8.1.32.2.88 BOM Reference

Location:
/components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/address/bom-ref

Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.32.2.89 Country

Location:
/components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/address/country

Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

8.1.32.2.90 Region

Location:
/components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/address/region

Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

8.1.32.2.91 Locality

Location:
/components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/address/locality

Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

8.1.32.2.92 Post Office Box Number

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/address/postOfficeBoxNumber

Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

8.1.32.2.93 Postal Code

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/address/postalCode

Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

8.1.32.2.94 Street Address

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/address/streetAddress

Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

8.1.32.2.95 Organization URL(s)

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/url

Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

8.1.32.2.96 Organizational Contact

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/contact
Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

8.1.32.2.97 Organizational Contact

Location:
/components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/contact/[]

Type: Object

Table 115 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

8.1.32.2.98 BOM Reference

Location:
/components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/contact/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.32.2.99 Name

Location:
/components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

8.1.32.2.100 Email Address

Location:
/components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

8.1.32.2.101 Phone

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

8.1.32.2.102 Organizational Contact

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/contact
Property: contact (Optional)

Type: Object

Description: The individual that is responsible for specific data governance role(s).

Table 116 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

8.1.32.2.103 BOM Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/contact/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.32.2.104 Name

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/contact/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

8.1.32.2.105 Email Address

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/contact/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

8.1.32.2.106 Phone

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/contact/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

8.1.32.2.107 Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/ref
Property: datasets (Optional)

Type: String

Description: References a data component by the components bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

8.1.32.2.108 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

8.1.32.2.109 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

8.1.32.2.110 Inputs

Location: /components/[]/modelCard/modelParameters/inputs
Property: inputs (Optional)

Type: Array

Description: The input format(s) of the model Each item of this array must be an Input and Output Parameters object.

8.1.32.2.111 Input and Output Parameters

Location: /components/[]/modelCard/modelParameters/inputs/[]

Type: Object

Table 117 – Properties for the inputs object

Property	Type	Requirement	Description
format	String	Optional	The data format for input/output to the model.

8.1.32.2.112 Input/Output Format

Location: /components/[]/modelCard/modelParameters/inputs/[]/format
Property: format (Optional)

Type: String

Description: The data format for input/output to the model.

Examples:

- string
- image
- time-series

8.1.32.2.113 Outputs

Location: /components/[]/modelCard/modelParameters/outputs
Property: outputs (Optional)

Type: Array

Description: The output format(s) from the model Each item of this array must be an Input and Output Parameters object.

8.1.32.2.114 Input and Output Parameters

Location: /components/[]/modelCard/modelParameters/outputs/[]

Type: Object

Table 118 – Properties for the outputs object

Property	Type	Requirement	Description
format	String	Optional	The data format for input/output to the model.

8.1.32.2.115 Input/Output Format

Location: /components/[]/modelCard/modelParameters/outputs/[]/format
Property: format (Optional)

Type: String

Description: The data format for input/output to the model.

Examples:

- string
- image
- time-series

8.1.32.3 Quantitative Analysis

Location: /components/[]/modelCard/quantitativeAnalysis
Property: quantitativeAnalysis (Optional)

Type: Object

Description: A quantitative analysis of the model

Table 119 – Properties for the quantitativeAnalysis object

Property	Type	Requirement	Description
performanceMetrics	Array	Optional	The model performance metrics being reported. Examples may include accuracy, F1 score, precision, top-3 error rates, MSC, etc.
graphics	Object	Optional	A collection of graphics that represent various measurements.

8.1.32.3.1 Performance Metrics

Location: /components/[]/modelCard/quantitativeAnalysis/performanceMetrics
Property: performanceMetrics (Optional)

Type: Array

Description: The model performance metrics being reported. Examples may include accuracy, F1 score, precision, top-3 error rates, MSC, etc. Each item of this array must be a Performance Metric object.

8.1.32.3.2 Performance Metric

Location: /components/[]/modelCard/quantitativeAnalysis/performanceMetrics/[]

Type: Object

Table 120 – Properties for the performanceMetrics object

Property	Type	Requirement	Description
type	String	Optional	The type of performance metric.
value	String	Optional	The value of the performance metric.
slice	String	Optional	The name of the slice this metric was computed on. By default, assume this metric is not sliced.
confidenceInterval	Object	Optional	The confidence interval of the metric.

8.1.32.3.3 Type

Location: /components/[]/modelCard/quantitativeAnalysis/performanceMetrics/[]/type
Property: type (Optional)

Type: String

Description: The type of performance metric.

8.1.32.3.4 Value

Location: /components/[]/modelCard/quantitativeAnalysis/performanceMetrics/[]/value
Property: value (Optional)

Type: String

Description: The value of the performance metric.

8.1.32.3.5 Slice

Location: /components/[]/modelCard/quantitativeAnalysis/performanceMetrics/[]/slice
Property: slice (Optional)

Type: String

Description: The name of the slice this metric was computed on. By default, assume this metric is not sliced.

8.1.32.3.6 Confidence Interval

Location: /components/[]/modelCard/quantitativeAnalysis/performanceMetrics/[]/confidenceInterval
Property: confidenceInterval (Optional)

Type: Object

Description: The confidence interval of the metric.

Table 121 – Properties for the confidenceInterval object

Property	Type	Requirement	Description
lowerBound	String	Optional	The lower bound of the confidence interval.
upperBound	String	Optional	The upper bound of the confidence interval.

8.1.32.3.7 Lower Bound

Location:
/components/[]/modelCard/quantitativeAnalysis/performanceMetrics/[]/confidenceInterval/lowerBound
Property: lowerBound (Optional)

Type: String

Description: The lower bound of the confidence interval.

8.1.32.3.8 Upper Bound

Location:
/components/[]/modelCard/quantitativeAnalysis/performanceMetrics/[]/confidenceInterval/upperBound
Property: upperBound (Optional)

Type: String

Description: The upper bound of the confidence interval.

8.1.32.3.9 Graphics Collection

Location: /components/[]/modelCard/quantitativeAnalysis/graphics
Property: graphics (Optional)

Type: Object

Description: A collection of graphics that represent various measurements.

Table 122 – Properties for the graphics object

Property	Type	Requirement	Description
description	String	Optional	A description of this collection of graphics.
collection	Array	Optional	A collection of graphics.

8.1.32.3.10 Description

Location: /components/[]/modelCard/quantitativeAnalysis/graphics/description
Property: description (Optional)

Type: String

Description: A description of this collection of graphics.

8.1.32.3.11 Collection

Location: /components/[]/modelCard/quantitativeAnalysis/graphics/collection
Property: collection (Optional)

Type: Array

Description: A collection of graphics. Each item of this array must be a Graphic object.

8.1.32.3.12 Graphic

Location: /components/[]/modelCard/quantitativeAnalysis/graphics/collection/[]

Type: Object

Table 123 – Properties for the collection object

Property	Type	Requirement	Description
name	String	Optional	The name of the graphic.
image	Object	Optional	The graphic (vector or raster). Base64 encoding must be specified for binary images.

8.1.32.3.13 Name

Location: /components/[]/modelCard/quantitativeAnalysis/graphics/collection/[]/name
Property: name (Optional)

Type: String

Description: The name of the graphic.

8.1.32.3.14 Graphic Image

Location: /components/[]/modelCard/quantitativeAnalysis/graphics/collection/[]/image
Property: image (Optional)

Type: Object

Description: The graphic (vector or raster). Base64 encoding must be specified for binary images.

Table 124 – Properties for the image object

Property	Type	Requirement	Description
contentType	String	Optional	Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.
encoding	String	Optional	Specifies the optional encoding the text is represented in.
content	String	Required	The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

8.1.32.3.15 Content-Type

Location: /components/[]/modelCard/quantitativeAnalysis/graphics/collection/[]/image/contentType
Property: contentType (Optional)

Type: String

Default Value: text/plain

Description: Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.

Examples:

- text/plain
- application/json
- image/png

8.1.32.3.16 Encoding

Location: /components/[]/modelCard/quantitativeAnalysis/graphics/collection/[]/image/encoding
Property: encoding (Optional)

Type: String

Description: Specifies the optional encoding the text is represented in.

Table 125 – Enumeration of possible values

Value	Description
base64	Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string.

8.1.32.3.17 Attachment Text

Location: /components/[]/modelCard/quantitativeAnalysis/graphics/collection/[]/image/content
Property: content (Required)

Type: String

Description: The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

8.1.32.4 Considerations

Location: /components/[]/modelCard/considerations
Property: considerations (Optional)

Type: Object

Description: What considerations should be taken into account regarding the model's construction, training, and application?

Table 126 – Properties for the considerations object

Property	Type	Requirement	Description
users	Array	Optional	Who are the intended users of the model?
useCases	Array	Optional	What are the intended use cases of the model?
technicalLimitations	Array	Optional	What are the known technical limitations of the model? E.g. What kind(s) of data should the model be expected not to perform well on? What are the factors that might degrade model performance?
performanceTradeoffs	Array	Optional	What are the known tradeoffs in accuracy/performance of the model?
ethicalConsiderations	Array	Optional	What are the ethical risks involved in the application of this model?
environmentalConsiderations	Object	Optional	What are the various environmental impacts the corresponding machine learning model has exhibited across its lifecycle?
fairnessAssessments	Array	Optional	How does the model affect groups at risk of being systematically disadvantaged? What are the harms and benefits to the various affected groups?

8.1.32.4.1 Users

Location: /components/[]/modelCard/considerations/users
Property: users (Optional)

Type: array (of String)

Description: Who are the intended users of the model? Each item of this array must be a string.

8.1.32.4.2 Use Cases

Location: /components/[]/modelCard/considerations/useCases
Property: useCases (Optional)

Type: array (of String)

Description: What are the intended use cases of the model? Each item of this array must be a string.

8.1.32.4.3 Technical Limitations

Location: /components/[]/modelCard/considerations/technicalLimitations

Property: technicalLimitations (Optional)

Type: array (of String)

Description: What are the known technical limitations of the model? E.g. What kind(s) of data should the model be expected not to perform well on? What are the factors that might degrade model performance? Each item of this array must be a string.

8.1.32.4.4 Performance Tradeoffs

Location: /components/[]/modelCard/considerations/performanceTradeoffs
Property: performanceTradeoffs (Optional)

Type: array (of String)

Description: What are the known tradeoffs in accuracy/performance of the model? Each item of this array must be a string.

8.1.32.4.5 Ethical Considerations

Location: /components/[]/modelCard/considerations/ethicalConsiderations
Property: ethicalConsiderations (Optional)

Type: Array

Description: What are the ethical risks involved in the application of this model? Each item of this array must be a Risk object.

8.1.32.4.6 Risk

Location: /components/[]/modelCard/considerations/ethicalConsiderations/[]

Type: Object

Table 127 – Properties for the ethicalConsiderations object

Property	Type	Requirement	Description
name	String	Optional	The name of the risk.
mitigationStrategy	String	Optional	Strategy used to address this risk.

8.1.32.4.7 Name

Location: /components/[]/modelCard/considerations/ethicalConsiderations/[]/name
Property: name (Optional)

Type: String

Description: The name of the risk.

8.1.32.4.8 Mitigation Strategy

Location: /components/[]/modelCard/considerations/ethicalConsiderations/[]/mitigationStrategy
Property: mitigationStrategy (Optional)

Type: String

Description: Strategy used to address this risk.

8.1.32.4.9 Environmental Considerations

Location: /components/[]/modelCard/considerations/environmentalConsiderations
Property: environmentalConsiderations (Optional)

Type: Object

Description: What are the various environmental impacts the corresponding machine learning model has exhibited across its lifecycle?

Table 128 – Properties for the environmentalConsiderations object

Property	Type	Requirement	Description
energyConsumptions	Array	Optional	Describes energy consumption information incurred for one or more component lifecycle activities.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

8.1.32.4.10 Energy Consumptions

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions
Property: energyConsumptions (Optional)

Type: Array

Description: Describes energy consumption information incurred for one or more component lifecycle activities. Each item of this array must be an Energy consumption object.

8.1.32.4.11 Energy consumption

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]

Type: Object

Description: Describes energy consumption information incurred for the specified lifecycle activity.

Table 129 – Properties for the energyConsumptions object

Property	Type	Requirement	Description
activity	String	Required	The type of activity that is part of a machine learning model development or operational lifecycle.
energyProviders	Array	Required	The provider(s) of the energy consumed by the associated model development lifecycle activity.
activityEnergyCost	Object	Required	The total energy cost associated with the model lifecycle activity.
co2CostEquivalent	Object	Optional	The CO2 cost (debit) equivalent to the total energy cost.
co2CostOffset	Object	Optional	The CO2 offset (credit) for the CO2 equivalent cost.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the

Property	Type	Requirement	Description
			general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

8.1.32.4.12 Activity

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/activity

Property: activity (Required)

Type: String

Description: The type of activity that is part of a machine learning model development or operational lifecycle.

Table 130 – Enumeration of possible values

Value	Description
design	A model design including problem framing, goal definition and algorithm selection.
data-collection	Model data acquisition including search, selection and transfer.
data-preparation	Model data preparation including data cleaning, labelling and conversion.
training	Model building, training and generalized tuning.
fine-tuning	Refining a trained model to produce desired outputs for a given problem space.
validation	Model validation including model output evaluation and testing.
deployment	Explicit model deployment to a target hosting infrastructure.
inference	Generating an output response from a hosted model from a set of inputs.
other	A lifecycle activity type whose description does not match currently defined values.

8.1.32.4.13 Energy Providers

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders

Property: energyProviders (Required)

Type: Array

Description: The provider(s) of the energy consumed by the associated model development lifecycle activity. Each item of this array must be an Energy Provider object.

8.1.32.4.14 Energy Provider

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]

Type: Object

Description: Describes the physical provider of energy used for model development or operations.

Table 131 – Properties for the energyProviders object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the energy provider elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
description	String	Optional	A description of the energy provider.
organization	Object	Required	The organization that provides energy.
energySource	String	Required	The energy source for the energy provider.
energyProvided	Object	Required	The energy provided by the energy source for an associated activity.
externalReferences	Array	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

8.1.32.4.15 BOM Reference

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/bom-ref

Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the energy provider elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.32.4.16 Description

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/description

Property: description (Optional)

Type: String

Description: A description of the energy provider.

8.1.32.4.17 Organization

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization

Property: organization (Required)

Type: Object

Description: The organization that provides energy.

Table 132 – Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

8.1.32.4.18 BOM Reference

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/bom-ref

Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.32.4.19 Organization Name

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/name

Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

8.1.32.4.20 Organization Address

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/address

Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 133 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

8.1.32.4.21 BOM Reference

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/address/bom-ref

Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.32.4.22 Country

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/address/country

Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

8.1.32.4.23 Region

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/address/region

Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

8.1.32.4.24 Locality

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/address/locality

Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

8.1.32.4.25 Post Office Box Number

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/address/postOfficeBoxNumber

Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

8.1.32.4.26 Postal Code

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/address/postalCode

Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

8.1.32.4.27 Street Address

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/address/streetAddress

Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

8.1.32.4.28 Organization URL(s)

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/url

Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

8.1.32.4.29 Organizational Contact

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/contact

Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

8.1.32.4.30 Organizational Contact

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/contact/[]

Type: Object

Table 134 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

8.1.32.4.31 BOM Reference

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/contact/[]/bom-ref

Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.32.4.32 Name

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/contact/[]/name

Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

8.1.32.4.33 Email Address

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/contact/[]/email

Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

8.1.32.4.34 Phone

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/contact/[]/phone

Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

8.1.32.4.35 Energy Source

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/energySource

Property: energySource (Required)

Type: String

Description: The energy source for the energy provider.

Table 135 – Enumeration of possible values

Value	Description
coal	Energy produced by types of coal.
oil	Petroleum products (primarily crude oil and its derivative fuel oils).
natural-gas	Hydrocarbon gas liquids (HGL) that occur as gasses at atmospheric pressure and as liquids under higher pressures including Natural gas (C5H12 and heavier), Ethane (C2H6), Propane (C3H8), etc.
nuclear	Energy produced from the cores of atoms (i.e., through nuclear fission or fusion).
wind	Energy produced from moving air.
solar	Energy produced from the sun (i.e., solar radiation).

Value	Description
geothermal	Energy produced from heat within the earth.
hydropower	Energy produced from flowing water.
biofuel	Liquid fuels produced from biomass feedstocks (i.e., organic materials such as plants or animals).
unknown	The energy source is unknown.
other	An energy source that is not listed.

8.1.32.4.36 Energy Provided

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/energyProvided

Property: energyProvided (Required)

Type: Object

Description: The energy provided by the energy source for an associated activity.

Table 136 – Properties for the energyProvided object

Property	Type	Requirement	Description
value	Number	Required	Quantity of energy.
unit	String	Required	Unit of energy.

8.1.32.4.37 Value

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/energyProvided/value

Property: value (Required)

Type: Number

Description: Quantity of energy.

8.1.32.4.38 Unit

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/energyProvided/unit

Property: unit (Required)

Type: String

Description: Unit of energy.

Table 137 – Enumeration of possible values

Value	Description
kWh	Kilowatt-hour (kWh) is the energy delivered by one kilowatt (kW) of power for one hour (h).

8.1.32.4.39 External References

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/externalReferences

Property: externalReferences (Optional)

Type: Array

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM. Each item of this array must be an External Reference object.

Reference: Refer to the external reference definition at /externalReferences/[]

8.1.32.4.40 External Reference

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/externalReferences/[]

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

8.1.32.4.41 Activity Energy Cost

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/activityEnergyCost

Property: activityEnergyCost (Required)

Type: Object

Description: The total energy cost associated with the model lifecycle activity.

Table 138 – Properties for the activityEnergyCost object

Property	Type	Requirement	Description
value	Number	Required	Quantity of energy.
unit	String	Required	Unit of energy.

8.1.32.4.42 Value

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/activityEnergyCost/value

Property: value (Required)

Type: Number

Description: Quantity of energy.

8.1.32.4.43 Unit

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/activityEnergyCost/unit

Property: unit (Required)

Type: String

Description: Unit of energy.

Table 139 – Enumeration of possible values

Value	Description
kWh	Kilowatt-hour (kWh) is the energy delivered by one kilowatt (kW) of power for one hour (h).

8.1.32.4.44 CO2 Equivalent Cost

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/co2CostEquivalent

Property: co2CostEquivalent (Optional)

Type: Object

Description: The CO2 cost (debit) equivalent to the total energy cost.

Table 140 – Properties for the co2CostEquivalent object

Property	Type	Requirement	Description
value	Number	Required	Quantity of carbon dioxide (CO2).
unit	String	Required	Unit of carbon dioxide (CO2).

8.1.32.4.45 Value

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/co2CostEquivalent/value

Property: value (Required)

Type: Number

Description: Quantity of carbon dioxide (CO2).

8.1.32.4.46 Unit

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/co2CostEquivalent/unit

Property: unit (Required)

Type: String

Description: Unit of carbon dioxide (CO2).

Table 141 – Enumeration of possible values

Value	Description
tCO2eq	Tons (t) of carbon dioxide (CO2) equivalent (eq).

8.1.32.4.47 CO2 Cost Offset

Location:
/components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/co2CostOffset

Property: co2CostOffset (Optional)

Type: Object

Description: The CO2 offset (credit) for the CO2 equivalent cost.

Table 142 – Properties for the co2CostOffset object

Property	Type	Requirement	Description
value	Number	Required	Quantity of carbon dioxide (CO2).
unit	String	Required	Unit of carbon dioxide (CO2).

8.1.32.4.48 Value

Location:
/components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/co2CostOffset/value

Property: value (Required)

Type: Number

Description: Quantity of carbon dioxide (CO2).

8.1.32.4.49 Unit

Location:
/components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/co2CostOffset/unit

Property: unit (Required)

Type: String

Description: Unit of carbon dioxide (CO2).

Table 143 – Enumeration of possible values

Value	Description
tCO2eq	Tons (t) of carbon dioxide (CO2) equivalent (eq).

8.1.32.4.50 Properties

Location:
/components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/properties

Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

8.1.32.4.51 Lightweight name-value pair

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 144 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

8.1.32.4.52 Name

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/properties/[]/name

Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

8.1.32.4.53 Value

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/properties/[]/value

Property: value (Optional)

Type: String

Description: The value of the property.

8.1.32.4.54 Properties

Location: /components/[]/modelCard/considerations/environmentalConsiderations/properties

Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different

values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

8.1.32.4.55 Lightweight name-value pair

Location: /components/[]/modelCard/considerations/environmentalConsiderations/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 145 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

8.1.32.4.56 Name

Location: /components/[]/modelCard/considerations/environmentalConsiderations/properties/[]/name

Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

8.1.32.4.57 Value

Location: /components/[]/modelCard/considerations/environmentalConsiderations/properties/[]/value

Property: value (Optional)

Type: String

Description: The value of the property.

8.1.32.4.58 Fairness Assessments

Location: /components/[]/modelCard/considerations/fairnessAssessments

Property: fairnessAssessments (Optional)

Type: Array

Description: How does the model affect groups at risk of being systematically disadvantaged? What are the harms and benefits to the various affected groups? Each item of this array must be a Fairness Assessment object.

8.1.32.4.59 Fairness Assessment

Location: /components/[]/modelCard/considerations/fairnessAssessments/[]

Type: Object

Description: Information about the benefits and harms of the model to an identified at risk group.

Table 146 – Properties for the fairnessAssessments object

Property	Type	Requirement	Description
groupAtRisk	String	Optional	The groups or individuals at risk of being systematically disadvantaged by the model.
benefits	String	Optional	Expected benefits to the identified groups.
harms	String	Optional	Expected harms to the identified groups.
mitigationStrategy	String	Optional	With respect to the benefits and harms outlined, please describe any mitigation strategy implemented.

8.1.32.4.60 Group at Risk

Location: /components/[]/modelCard/considerations/fairnessAssessments/[]/groupAtRisk
Property: groupAtRisk (Optional)

Type: String

Description: The groups or individuals at risk of being systematically disadvantaged by the model.

8.1.32.4.61 Benefits

Location: /components/[]/modelCard/considerations/fairnessAssessments/[]/benefits
Property: benefits (Optional)

Type: String

Description: Expected benefits to the identified groups.

8.1.32.4.62 Harms

Location: /components/[]/modelCard/considerations/fairnessAssessments/[]/harms
Property: harms (Optional)

Type: String

Description: Expected harms to the identified groups.

8.1.32.4.63 Mitigation Strategy

Location: /components/[]/modelCard/considerations/fairnessAssessments/[]/mitigationStrategy
Property: mitigationStrategy (Optional)

Type: String

Description: With respect to the benefits and harms outlined, please describe any mitigation strategy implemented.

8.1.32.5 Properties

Location: /components/[]/modelCard/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX

Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

8.1.32.5.1 Lightweight name-value pair

Location: /components/[]/modelCard/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 147 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

8.1.32.5.2 Name

Location: /components/[]/modelCard/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

8.1.32.5.3 Value

Location: /components/[]/modelCard/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

8.1.33 Data

Location: /components/[]/data
Property: data (Optional)

Type: Array

Description: This object SHOULD be specified for any component of type data and must not be specified for other component types.

Location: /components/[]/data/[]

Type: Object

Table 148 – Properties for the data object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the dataset elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
type	String	Required	The general theme or subject matter of the data being specified.
name	String	Optional	The name of the dataset.
contents	Object	Optional	The contents or references to the contents of the data being described.
classification	String	Optional	Data classification tags data according to its type, sensitivity, and value if altered, stolen, or destroyed.
sensitiveData	Array	Optional	A description of any sensitive data in a dataset.
graphics	Object	Optional	A collection of graphics that represent various measurements.
description	String	Optional	A description of the dataset. Can describe size of dataset, whether it's used for source code, training, testing, or validation, etc.
governance	Object	Optional	Data governance captures information regarding data ownership, stewardship, and custodianship, providing insights into the individuals or entities responsible for managing, overseeing, and safeguarding the data throughout its lifecycle.

8.1.33.1.1 BOM Reference

Location: /components/[]/data/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the dataset elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.33.1.2 Type of Data

Location: /components/[]/data/[]/type
Property: type (Required)

Type: String

Description: The general theme or subject matter of the data being specified.

Table 149 – Enumeration of possible values

Value	Description
source-code	Any type of code, code snippet, or data-as-code.
configuration	Parameters or settings that may be used by other components.
dataset	A collection of data.
definition	Data that can be used to create new instances of what the definition defines.
other	Any other type of data that does not fit into existing definitions.

8.1.33.1.3 Dataset Name

Location: /components/[]/data/[]/name
Property: name (Optional)

Type: String

Description: The name of the dataset.

8.1.33.1.4 Data Contents

Location: /components/[]/data/[]/contents
Property: contents (Optional)

Type: Object

Description: The contents or references to the contents of the data being described.

Table 150 – Properties for the contents object

Property	Type	Requirement	Description
attachment	Object	Optional	An optional way to include textual or encoded data.
url	String	Optional	The URL to where the data can be retrieved.
properties	Array	Optional	Provides the ability to document name-value parameters used for configuration.

8.1.33.1.5 Data Attachment

Location: /components/[]/data/[]/contents/attachment
Property: attachment (Optional)

Type: Object

Description: An optional way to include textual or encoded data.

Table 151 – Properties for the attachment object

Property	Type	Requirement	Description
contentType	String	Optional	Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.
encoding	String	Optional	Specifies the optional encoding the text is represented in.
content	String	Required	The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

8.1.33.1.6 Content-Type

Location: /components/[]/data/[]/contents/attachment/contentType
Property: contentType (Optional)

Type: String

Default Value: text/plain

Description: Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.

Examples:

- text/plain
- application/json
- image/png

8.1.33.1.7 Encoding

Location: /components/[]/data/[]/contents/attachment/encoding
Property: encoding (Optional)

Type: String

Description: Specifies the optional encoding the text is represented in.

Table 152 – Enumeration of possible values

Value	Description
base64	Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string.

8.1.33.1.8 Attachment Text

Location: /components/[]/data/[]/contents/attachment/content
Property: content (Required)

Type: String

Description: The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

8.1.33.1.9 Data URL

Location: /components/[]/data/[]/contents/url
Property: url (Optional)

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL to where the data can be retrieved.

8.1.33.1.10 Configuration Properties

Location: /components/[]/data/[]/contents/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document name-value parameters used for configuration. Each item of this array must be a Lightweight name-value pair object.

8.1.33.1.11 Lightweight name-value pair

Location: /components/[]/data/[]/contents/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 153 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

8.1.33.1.12 Name

Location: /components/[]/data/[]/contents/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

8.1.33.1.13 Value

Location: /components/[]/data/[]/contents/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

8.1.33.1.14 Data Classification

Location: /components/[]/data/[]/classification
Property: classification (Optional)

Type: String

Description: Data classification tags data according to its type, sensitivity, and value if altered, stolen, or destroyed.

8.1.33.1.15 Sensitive Data

Location: /components/[]/data/[]/sensitiveData
Property: sensitiveData (Optional)

Type: array (of String)

Description: A description of any sensitive data in a dataset. Each item of this array must be a string.

8.1.33.1.16 Graphics Collection

Location: /components/[]/data/[]/graphics
Property: graphics (Optional)

Type: Object

Description: A collection of graphics that represent various measurements.

Table 154 – Properties for the graphics object

Property	Type	Requirement	Description
description	String	Optional	A description of this collection of graphics.
collection	Array	Optional	A collection of graphics.

8.1.33.1.17 Description

Location: /components/[]/data/[]/graphics/description
Property: description (Optional)

Type: String

Description: A description of this collection of graphics.

8.1.33.1.18 Collection

Location: /components/[]/data/[]/graphics/collection
Property: collection (Optional)

Type: Array

Description: A collection of graphics. Each item of this array must be a Graphic object.

8.1.33.1.19 Graphic

Location: /components/[]/data/[]/graphics/collection/[]

Type: Object

Table 155 – Properties for the collection object

Property	Type	Requirement	Description
name	String	Optional	The name of the graphic.
image	Object	Optional	The graphic (vector or raster). Base64 encoding must be specified for binary images.

8.1.33.1.20 Name

Location: /components/[]/data/[]/graphics/collection/[]/name
Property: name (Optional)

Type: String

Description: The name of the graphic.

8.1.33.1.21 Graphic Image

Location: /components/[]/data/[]/graphics/collection/[]/image
Property: image (Optional)

Type: Object

Description: The graphic (vector or raster). Base64 encoding must be specified for binary images.

Table 156 – Properties for the image object

Property	Type	Requirement	Description
contentType	String	Optional	Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.
encoding	String	Optional	Specifies the optional encoding the text is represented in.
content	String	Required	The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

8.1.33.1.22 Content-Type

Location: /components/[]/data/[]/graphics/collection/[]/image/contentType
Property: contentType (Optional)

Type: String

Default Value: text/plain

Description: Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.

Examples:

- text/plain

- application/json
- image/png

8.1.33.1.23 Encoding

Location: /components/[]/data/[]/graphics/collection/[]/image/encoding
Property: encoding (Optional)

Type: String

Description: Specifies the optional encoding the text is represented in.

Table 157 – Enumeration of possible values

Value	Description
base64	Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string.

8.1.33.1.24 Attachment Text

Location: /components/[]/data/[]/graphics/collection/[]/image/content
Property: content (Required)

Type: String

Description: The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

8.1.33.1.25 Dataset Description

Location: /components/[]/data/[]/description
Property: description (Optional)

Type: String

Description: A description of the dataset. Can describe size of dataset, whether it's used for source code, training, testing, or validation, etc.

8.1.33.1.26 Data Governance

Location: /components/[]/data/[]/governance
Property: governance (Optional)

Type: Object

Description: Data governance captures information regarding data ownership, stewardship, and custodianship, providing insights into the individuals or entities responsible for managing, overseeing, and safeguarding the data throughout its lifecycle.

Table 158 – Properties for the governance object

Property	Type	Requirement	Description
custodians	Array	Optional	Data custodians are responsible for the safe custody, transport, and storage of data.
stewards	Array	Optional	Data stewards are responsible for data content, context, and associated business rules.
owners	Array	Optional	Data owners are concerned with risk and appropriate access to data.

8.1.33.1.27 Data Custodians

Location: /components/[]/data/[]/governance/custodians
Property: custodians (Optional)

Type: Array

Description: Data custodians are responsible for the safe custody, transport, and storage of data.

Location: /components/[]/data/[]/governance/custodians/[]

Type: Object

Table 159 – Properties for the custodians object

Property	Type	Requirement	Description
organization	Object	Optional	The organization that is responsible for specific data governance role(s).
contact	Object	Optional	The individual that is responsible for specific data governance role(s).

8.1.33.1.28 Organization

Location: /components/[]/data/[]/governance/custodians/[]/organization
Property: organization (Optional)

Type: Object

Description: The organization that is responsible for specific data governance role(s).

Table 160 – Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

8.1.33.1.29 BOM Reference

Location: /components/[]/data/[]/governance/custodians/[]/organization/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.33.1.30 Organization Name

Location: /components/[]/data/[]/governance/custodians/[]/organization/name

Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

8.1.33.1.31 Organization Address

Location: /components/[]/data/[]/governance/custodians/[]/organization/address
Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 161 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

8.1.33.1.32 BOM Reference

Location: /components/[]/data/[]/governance/custodians/[]/organization/address/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.33.1.33 Country

Location: /components/[]/data/[]/governance/custodians/[]/organization/address/country
Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

8.1.33.1.34 Region

Location: /components/[]/data/[]/governance/custodians/[]/organization/address/region
Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

8.1.33.1.35 Locality

Location: /components/[]/data/[]/governance/custodians/[]/organization/address/locality
Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

8.1.33.1.36 Post Office Box Number

Location: /components/[]/data/[]/governance/custodians/[]/organization/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

8.1.33.1.37 Postal Code

Location: /components/[]/data/[]/governance/custodians/[]/organization/address/postalCode
Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

8.1.33.1.38 Street Address

Location: /components/[]/data/[]/governance/custodians/[]/organization/address/streetAddress
Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

8.1.33.1.39 Organization URL(s)

Location: /components/[]/data/[]/governance/custodians/[]/organization/url
Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

8.1.33.1.40 Organizational Contact

Location: /components/[]/data/[]/governance/custodians/[]/organization/contact
Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

8.1.33.1.41 Organizational Contact

Location: /components/[]/data/[]/governance/custodians/[]/organization/contact/[]

Type: Object

Table 162 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

8.1.33.1.42 BOM Reference

Location: /components/[]/data/[]/governance/custodians/[]/organization/contact/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.33.1.43 Name

Location: /components/[]/data/[]/governance/custodians/[]/organization/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

8.1.33.1.44 Email Address

Location: /components/[]/data/[]/governance/custodians/[]/organization/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

8.1.33.1.45 Phone

Location: /components/[]/data/[]/governance/custodians/[]/organization/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

8.1.33.1.46 Organizational Contact

Location: /components/[]/data/[]/governance/custodians/[]/contact
Property: contact (Optional)

Type: Object

Description: The individual that is responsible for specific data governance role(s).

Table 163 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

8.1.33.1.47 BOM Reference

Location: /components/[]/data/[]/governance/custodians/[]/contact/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.33.1.48 Name

Location: /components/[]/data/[]/governance/custodians/[]/contact/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

8.1.33.1.49 Email Address

Location: /components/[]/data/[]/governance/custodians/[]/contact/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

8.1.33.1.50 Phone

Location: /components/[]/data/[]/governance/custodians/[]/contact/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

8.1.33.1.51 Data Stewards

Location: /components/[]/data/[]/governance/stewards
Property: stewards (Optional)

Type: Array

Description: Data stewards are responsible for data content, context, and associated business rules.

Location: /components/[]/data/[]/governance/stewards/[]

Type: Object

Table 164 – Properties for the stewards object

Property	Type	Requirement	Description
organization	Object	Optional	The organization that is responsible for specific data governance role(s).
contact	Object	Optional	The individual that is responsible for specific data governance role(s).

8.1.33.1.52 Organization

Location: /components/[]/data/[]/governance/stewards/[]/organization
Property: organization (Optional)

Type: Object

Description: The organization that is responsible for specific data governance role(s).

Table 165 – Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

8.1.33.1.53 BOM Reference

Location: /components/[]/data/[]/governance/stewards/[]/organization/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.33.1.54 Organization Name

Location: /components/[]/data/[]/governance/stewards/[]/organization/name
Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

8.1.33.1.55 Organization Address

Location: /components/[]/data/[]/governance/stewards/[]/organization/address
Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 166 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

8.1.33.1.56 BOM Reference

Location: /components/[]/data/[]/governance/stewards/[]/organization/address/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.33.1.57 Country

Location: /components/[]/data/[]/governance/stewards/[]/organization/address/country
Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

8.1.33.1.58 Region

Location: /components/[]/data/[]/governance/stewards/[]/organization/address/region
Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

8.1.33.1.59 Locality

Location: /components/[]/data/[]/governance/stewards/[]/organization/address/locality
Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

8.1.33.1.60 Post Office Box Number

Location: /components/[]/data/[]/governance/stewards/[]/organization/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

8.1.33.1.61 Postal Code

Location: /components/[]/data/[]/governance/stewards/[]/organization/address/postalCode
Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

8.1.33.1.62 Street Address

Location: /components/[]/data/[]/governance/stewards/[]/organization/address/streetAddress
Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

8.1.33.1.63 Organization URL(s)

Location: /components/[]/data/[]/governance/stewards/[]/organization/url
Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

8.1.33.1.64 Organizational Contact

Location: /components/[]/data/[]/governance/stewards/[]/organization/contact
Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

8.1.33.1.65 Organizational Contact

Location: /components/[]/data/[]/governance/stewards/[]/organization/contact/[]

Type: Object

Table 167 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

8.1.33.1.66 BOM Reference

Location: /components/[]/data/[]/governance/stewards/[]/organization/contact/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.33.1.67 Name

Location: /components/[]/data/[]/governance/stewards/[]/organization/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

8.1.33.1.68 Email Address

Location: /components/[]/data/[]/governance/stewards/[]/organization/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

8.1.33.1.69 Phone

Location: /components/[]/data/[]/governance/stewards/[]/organization/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

8.1.33.1.70 Organizational Contact

Location: /components/[]/data/[]/governance/stewards/[]/contact
Property: contact (Optional)

Type: Object

Description: The individual that is responsible for specific data governance role(s).

Table 168 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

8.1.33.1.71 BOM Reference

Location: /components/[]/data/[]/governance/stewards/[]/contact/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.33.1.72 Name

Location: /components/[]/data/[]/governance/stewards/[]/contact/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

8.1.33.1.73 Email Address

Location: /components/[]/data/[]/governance/stewards/[]/contact/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

8.1.33.1.74 Phone

Location: /components/[]/data/[]/governance/stewards/[]/contact/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

8.1.33.1.75 Data Owners

Location: /components/[]/data/[]/governance/owners
Property: owners (Optional)

Type: Array

Description: Data owners are concerned with risk and appropriate access to data.

Location: /components/[]/data/[]/governance/owners/[]

Type: Object

Table 169 – Properties for the owners object

Property	Type	Requirement	Description
organization	Object	Optional	The organization that is responsible for specific data governance role(s).
contact	Object	Optional	The individual that is responsible for specific data governance role(s).

8.1.33.1.76 Organization

Location: /components/[]/data/[]/governance/owners/[]/organization
Property: organization (Optional)

Type: Object

Description: The organization that is responsible for specific data governance role(s).

Table 170 – Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

8.1.33.1.77 BOM Reference

Location: /components/[]/data/[]/governance/owners/[]/organization/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.33.1.78 Organization Name

Location: /components/[]/data/[]/governance/owners/[]/organization/name
Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

8.1.33.1.79 Organization Address

Location: /components/[]/data/[]/governance/owners/[]/organization/address
Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 171 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

8.1.33.1.80 BOM Reference

Location: /components/[]/data/[]/governance/owners/[]/organization/address/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.33.1.81 Country

Location: /components/[]/data/[]/governance/owners/[]/organization/address/country
Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

8.1.33.1.82 Region

Location: /components/[]/data/[]/governance/owners/[]/organization/address/region
Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

8.1.33.1.83 Locality

Location: /components/[]/data/[]/governance/owners/[]/organization/address/locality
Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

8.1.33.1.84 Post Office Box Number

Location: /components/[]/data/[]/governance/owners/[]/organization/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

8.1.33.1.85 Postal Code

Location: /components/[]/data/[]/governance/owners/[]/organization/address/postalCode
Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

8.1.33.1.86 Street Address

Location: /components/[]/data/[]/governance/owners/[]/organization/address/streetAddress
Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

8.1.33.1.87 Organization URL(s)

Location: /components/[]/data/[]/governance/owners/[]/organization/url
Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

8.1.33.1.88 Organizational Contact

Location: /components/[]/data/[]/governance/owners/[]/organization/contact
Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

8.1.33.1.89 Organizational Contact

Location: /components/[]/data/[]/governance/owners/[]/organization/contact/[]

Type: Object

Table 172 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

8.1.33.1.90 BOM Reference

Location: /components/[]/data/[]/governance/owners/[]/organization/contact/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.33.1.91 Name

Location: /components/[]/data/[]/governance/owners/[]/organization/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

8.1.33.1.92 Email Address

Location: /components/[]/data/[]/governance/owners/[]/organization/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

8.1.33.1.93 Phone

Location: /components/[]/data/[]/governance/owners/[]/organization/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

8.1.33.1.94 Organizational Contact

Location: /components/[]/data/[]/governance/owners/[]/contact
Property: contact (Optional)

Type: Object

Description: The individual that is responsible for specific data governance role(s).

Table 173 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

8.1.33.1.95 BOM Reference

Location: /components/[]/data/[]/governance/owners/[]/contact/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

8.1.33.1.96 Name

Location: /components/[]/data/[]/governance/owners/[]/contact/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

8.1.33.1.97 Email Address

Location: /components/[]/data/[]/governance/owners/[]/contact/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

8.1.33.1.98 Phone

Location: /components/[]/data/[]/governance/owners/[]/contact/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

8.1.34 Cryptographic Properties

Location: /components/[]/cryptoProperties
Property: cryptoProperties (Optional)

Type: Object

Description: Cryptographic assets have properties that uniquely define them and that make them actionable for further reasoning. As an example, it makes a difference if one knows the algorithm family (e.g. AES) or the specific variant or instantiation (e.g. AES-128-GCM). This is because the security level and the algorithm primitive (authenticated encryption) are only defined by the definition of the algorithm variant. The presence of a weak cryptographic algorithm like SHA1 vs. HMAC-SHA1 also makes a difference.

Table 174 – Properties for the cryptoProperties object

Property	Type	Requirement	Description
assetType	String	Required	Cryptographic assets occur in several forms. Algorithms and protocols are most commonly implemented in specialized cryptographic libraries. They may, however, also be 'hardcoded' in software components. Certificates and related cryptographic material like keys, tokens, secrets or passwords are other cryptographic assets to be modelled.
algorithmProperties	Object	Optional	Additional properties specific to a cryptographic algorithm.
certificateProperties	Object	Optional	Properties for cryptographic assets of asset type 'certificate'
relatedCryptoMaterialProperties	Object	Optional	Properties for cryptographic assets of asset type: related-crypto-material
protocolProperties	Object	Optional	Properties specific to cryptographic assets of type: protocol.
oid	String	Optional	The object identifier (OID) of the cryptographic asset.

8.1.34.1 Asset Type

Location: /components/[]/cryptoProperties/assetType
Property: assetType (Required)

Type: String

Description: Cryptographic assets occur in several forms. Algorithms and protocols are most commonly implemented in specialized cryptographic libraries. They may, however, also be 'hardcoded' in software components. Certificates and related cryptographic material like keys, tokens, secrets or passwords are other cryptographic assets to be modeled.

Table 175 – Enumeration of possible values

Value	Description
algorithm	Mathematical function commonly used for data encryption, authentication, and digital signatures.
certificate	An electronic document that is used to provide the identity or validate a public key.
protocol	A set of rules and guidelines that govern the behaviour and communication with each other.
related-crypto-material	Other cryptographic assets related to algorithms, certificates, and protocols such as keys and tokens.

8.1.34.2 Algorithm Properties

Location: /components/[]/cryptoProperties/algorithmProperties
Property: algorithmProperties (Optional)

Type: Object

Description: Additional properties specific to a cryptographic algorithm.

Table 176 – Properties for the algorithmProperties object

Property	Type	Requirement	Description
primitive	String	Optional	Cryptographic building blocks used in higher-level cryptographic systems and protocols. Primitives represent different cryptographic routines: deterministic random bit generators (drbg, e.g. CTR_DRBG from NIST SP800-90A-r1), message authentication codes (mac, e.g. HMAC-SHA-256), blockciphers (e.g. AES), streamciphers (e.g. Salsa20), signatures (e.g. ECDSA), hash functions (e.g. SHA-256), public-key encryption schemes (pke, e.g. RSA), extended output functions (xof, e.g. SHAKE256), key derivation functions (e.g. pbkdf2), key agreement algorithms (e.g. ECDH), key encapsulation mechanisms (e.g. ML-KEM), authenticated encryption (ae, e.g. AES-GCM) and the combination of multiple algorithms (combiner, e.g. SP800-56Cr2).
parameterSetIdentifier	String	Optional	An identifier for the parameter set of the cryptographic algorithm. Examples: in AES128, '128' identifies the key length in bits, in SHA256, '256' identifies the digest length, '128' in SHAKE128 identifies its maximum security level in bits, and 'SHA2-128s' identifies a parameter set used in SLH-DSA (FIPS205).
curve	String	Optional	The specific underlying Elliptic Curve (EC) definition employed which is an indicator of the level of security strength, performance and complexity. Absent an authoritative source of curve names, CycloneDX recommends using curve names as defined at https://neuromancer.sk/std/, the source of which can be found at https://github.com/J08nY/std-curves.
executionEnvironment	String	Optional	The target and execution environment in which the algorithm is implemented in.
implementationPlatform	String	Optional	The target platform for which the algorithm is implemented. The implementation can be 'generic', running on any platform or for a specific platform.
certificationLevel	Array	Optional	The certification that the implementation of the cryptographic algorithm has received, if any. Certifications include revisions and levels of FIPS 140 or Common Criteria of different Extended Assurance Levels (CC-EAL).
mode	String	Optional	The mode of operation in which the cryptographic algorithm (block cipher) is used.
padding	String	Optional	The padding scheme that is used for the cryptographic algorithm.
cryptoFunctions	Array	Optional	The cryptographic functions implemented by the cryptographic algorithm.

Property	Type	Requirement	Description
classicalSecurityLevel	Integer	Optional	The classical security level that a cryptographic algorithm provides (in bits).
nistQuantumSecurityLevel	Integer	Optional	The NIST security strength category as defined in https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/security-(evaluation-criteria). A value of 0 indicates that none of the categories are met.

8.1.34.2.1 primitive

Location: /components/[]/cryptoProperties/algorithmProperties/primitive
Property: primitive (Optional)

Type: String

Description: Cryptographic building blocks used in higher-level cryptographic systems and protocols. Primitives represent different cryptographic routines: deterministic random bit generators (drbg, e.g. CTR_DRBG from NIST SP800-90A-r1), message authentication codes (mac, e.g. HMAC-SHA-256), blockciphers (e.g. AES), streamciphers (e.g. Salsa20), signatures (e.g. ECDSA), hash functions (e.g. SHA-256), public-key encryption schemes (pke, e.g. RSA), extended output functions (xof, e.g. SHAKE256), key derivation functions (e.g. pbkdf2), key agreement algorithms (e.g. ECDH), key encapsulation mechanisms (e.g. ML-KEM), authenticated encryption (ae, e.g. AES-GCM) and the combination of multiple algorithms (combiner, e.g. SP800-56Cr2).

Table 177 – Enumeration of possible values

Value	Description
drbg	Deterministic Random Bit Generator (DRBG) is a type of pseudorandom number generator designed to produce a sequence of bits from an initial seed value. DRBGs are commonly used in cryptographic applications where reproducibility of random values is important.
mac	In cryptography, a Message Authentication Code (MAC) is information used for authenticating and integrity-checking a message.
block-cipher	A block cipher is a symmetric key algorithm that operates on fixed-size blocks of data. It encrypts or decrypts the data in block units, providing confidentiality. Block ciphers are widely used in various cryptographic modes and protocols for secure data transmission.
stream-cipher	A stream cipher is a symmetric key cipher where plaintext digits are combined with a pseudorandom cipher digit stream (keystream).
signature	In cryptography, a signature is a digital representation of a message or data that proves its origin, identity, and integrity. Digital signatures are generated using cryptographic algorithms and are widely used for authentication and verification in secure communication.
hash	A hash function is a mathematical algorithm that takes an input (or 'message') and produces a fixed-size string of characters, which is typically a hash value. Hash functions are commonly used in various cryptographic applications, including data integrity verification and password hashing.
pke	Public Key Encryption (PKE) is a type of encryption that uses a pair of public and private keys for secure communication. The public key is used for encryption, while the private key is used for decryption. PKE is a fundamental component of public-key cryptography.

Value	Description
xof	An XOF is an extendable output function that can take arbitrary input and creates a stream of output, up to a limit determined by the size of the internal state of the hash function that underlies the XOF.
kdf	A Key Derivation Function (KDF) derives key material from another source of entropy while preserving the entropy of the input.
key-agree	In cryptography, a key-agreement is a protocol whereby two or more parties agree on a cryptographic key in such a way that both influence the outcome.
kem	A Key Encapsulation Mechanism (KEM) algorithm is a mechanism for transporting random keying material to a recipient using the recipient's public key.
ae	Authenticated Encryption (AE) is a cryptographic process that provides both confidentiality and data integrity. It ensures that the encrypted data has not been tampered with and comes from a legitimate source. AE is commonly used in secure communication protocols.
combiner	A combiner aggregates many candidates for a cryptographic primitive and generates a new candidate for the same primitive.
other	Another primitive type.
unknown	The primitive is not known.

8.1.34.2.2 Parameter Set Identifier

Location: /components/[]/cryptoProperties/algorithmProperties/parameterSetIdentifier
Property: parameterSetIdentifier (Optional)

Type: String

Description: An identifier for the parameter set of the cryptographic algorithm. Examples: in AES128, '128' identifies the key length in bits, in SHA256, '256' identifies the digest length, '128' in SHAKE128 identifies its maximum security level in bits, and 'SHA2-128s' identifies a parameter set used in SLH-DSA (FIPS205).

8.1.34.2.3 Elliptic Curve

Location: /components/[]/cryptoProperties/algorithmProperties/curve
Property: curve (Optional)

Type: String

Description: The specific underlying Elliptic Curve (EC) definition employed which is an indicator of the level of security strength, performance and complexity. Absent an authoritative source of curve names, CycloneDX recommends using curve names as defined at https://neuromancer.sk/std/, the source of which can be found at https://github.com/J08nY/std-curves.

8.1.34.2.4 Execution Environment

Location: /components/[]/cryptoProperties/algorithmProperties/executionEnvironment
Property: executionEnvironment (Optional)

Type: String

Description: The target and execution environment in which the algorithm is implemented in.

Table 178 – Enumeration of possible values

Value	Description
software-plain-ram	A software implementation running in plain unencrypted RAM.
software-encrypted-ram	A software implementation running in encrypted RAM.
software-tee	A software implementation running in a trusted execution environment.
hardware	A hardware implementation.
other	Another implementation environment.
unknown	The execution environment is not known.

8.1.34.2.5 Implementation platform

Location: /components/[]/cryptoProperties/algorithmProperties/implementationPlatform
Property: implementationPlatform (Optional)

Type: String

Description: The target platform for which the algorithm is implemented. The implementation can be 'generic', running on any platform or for a specific platform.

Enumeration: Must be one of:

- generic
- x86_32
- x86_64
- armv7-a
- armv7-m
- armv8-a
- armv8-m
- armv9-a
- armv9-m
- s390x
- ppc64
- ppc64le
- other
- unknown

8.1.34.2.6 Certification Level

Location: /components/[]/cryptoProperties/algorithmProperties/certificationLevel
Property: certificationLevel (Optional)

Type: array (of String)

Description: The certification that the implementation of the cryptographic algorithm has received, if any. Certifications include revisions and levels of FIPS 140 or Common Criteria of different Extended Assurance Levels (CC-EAL). Each item of this array must be a string.

Table 179 – Enumeration of possible values

Value	Description
none	No certification obtained
fips140-1-l1	FIPS 140-1 Level 1
fips140-1-l2	FIPS 140-1 Level 2
fips140-1-l3	FIPS 140-1 Level 3
fips140-1-l4	FIPS 140-1 Level 4
fips140-2-l1	FIPS 140-2 Level 1
fips140-2-l2	FIPS 140-2 Level 2
fips140-2-l3	FIPS 140-2 Level 3
fips140-2-l4	FIPS 140-2 Level 4
fips140-3-l1	FIPS 140-3 Level 1
fips140-3-l2	FIPS 140-3 Level 2
fips140-3-l3	FIPS 140-3 Level 3
fips140-3-l4	FIPS 140-3 Level 4
cc-eal1	Common Criteria - Evaluation Assurance Level 1
cc-eal1+	Common Criteria - Evaluation Assurance Level 1 (Augmented)
cc-eal2	Common Criteria - Evaluation Assurance Level 2
cc-eal2+	Common Criteria - Evaluation Assurance Level 2 (Augmented)
cc-eal3	Common Criteria - Evaluation Assurance Level 3
cc-eal3+	Common Criteria - Evaluation Assurance Level 3 (Augmented)
cc-eal4	Common Criteria - Evaluation Assurance Level 4
cc-eal4+	Common Criteria - Evaluation Assurance Level 4 (Augmented)
cc-eal5	Common Criteria - Evaluation Assurance Level 5
cc-eal5+	Common Criteria - Evaluation Assurance Level 5 (Augmented)

Value	Description
cc-eal6	Common Criteria - Evaluation Assurance Level 6
cc-eal6+	Common Criteria - Evaluation Assurance Level 6 (Augmented)
cc-eal7	Common Criteria - Evaluation Assurance Level 7
cc-eal7+	Common Criteria - Evaluation Assurance Level 7 (Augmented)
other	Another certification
unknown	The certification level is not known

8.1.34.2.7 Mode

Location: /components/[]/cryptoProperties/algorithmProperties/mode
Property: mode (Optional)

Type: String

Description: The mode of operation in which the cryptographic algorithm (block cipher) is used.

Table 180 – Enumeration of possible values

Value	Description
cbc	Cipher block chaining
ecb	Electronic codebook
ccm	Counter with cipher block chaining message authentication code
gcm	Galois/counter
cfb	Cipher feedback
ofb	Output feedback
ctr	Counter
other	Another mode of operation
unknown	The mode of operation is not known

8.1.34.2.8 Padding

Location: /components/[]/cryptoProperties/algorithmProperties/padding
Property: padding (Optional)

Type: String

Description: The padding scheme that is used for the cryptographic algorithm.

Table 181 – Enumeration of possible values

Value	Description
pkcs5	Public Key Cryptography Standard: Password-Based Cryptography
pkcs7	Public Key Cryptography Standard: Cryptographic Message Syntax
pkcs1v15	Public Key Cryptography Standard: RSA Cryptography v1.5
oaep	Optimal asymmetric encryption padding
raw	Raw
other	Another padding scheme
unknown	The padding scheme is not known

8.1.34.2.9 Cryptographic functions

Location: /components/[]/cryptoProperties/algorithmProperties/cryptoFunctions
Property: cryptoFunctions (Optional)

Type: array (of String)

Description: The cryptographic functions implemented by the cryptographic algorithm. Each item of this array must be a string.

Enumeration: Must be one of:

- generate
- keygen
- encrypt
- decrypt
- digest
- tag
- keyderive
- sign
- verify
- encapsulate
- decapsulate
- other
- unknown

8.1.34.2.10 classical security level

Location: /components/[]/cryptoProperties/algorithmProperties/classicalSecurityLevel
Property: classicalSecurityLevel (Optional)

Type: Integer

Description: The classical security level that a cryptographic algorithm provides (in bits).

8.1.34.2.11 NIST security strength category

Location: /components/[]/cryptoProperties/algorithmProperties/nistQuantumSecurityLevel
Property: nistQuantumSecurityLevel (Optional)

Type: Integer

Maximum Value: 6

Description: The NIST security strength category as defined in https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/security-(evaluation-criteria). A value of 0 indicates that none of the categories are met.

8.1.34.3 Certificate Properties

Location: /components/[]/cryptoProperties/certificateProperties
Property: certificateProperties (Optional)

Type: Object

Description: Properties for cryptographic assets of asset type 'certificate'

Table 182 – Properties for the certificateProperties object

Property	Type	Requirement	Description
subjectName	String	Optional	The subject name for the certificate
issuerName	String	Optional	The issuer name for the certificate
notValidBefore	String	Optional	The date and time according to ISO-8601 standard from which the certificate is valid
notValidAfter	String	Optional	The date and time according to ISO-8601 standard from which the certificate is not valid anymore
signatureAlgorithmRef	String	Optional	The bom-ref to signature algorithm used by the certificate
subjectPublicKeyRef	String	Optional	The bom-ref to the public key of the subject
certificateFormat	String	Optional	The format of the certificate
certificateExtension	String	Optional	The file extension of the certificate

8.1.34.3.1 Subject Name

Location: /components/[]/cryptoProperties/certificateProperties/subjectName
Property: subjectName (Optional)

Type: String

Description: The subject name for the certificate

8.1.34.3.2 Issuer Name

Location: /components/[]/cryptoProperties/certificateProperties/issuerName
Property: issuerName (Optional)

Type: String

Description: The issuer name for the certificate

8.1.34.3.3 Not Valid Before

Location: /components/[]/cryptoProperties/certificateProperties/notValidBefore
Property: notValidBefore (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time according to ISO-8601 standard from which the certificate is valid

8.1.34.3.4 Not Valid After

Location: /components/[]/cryptoProperties/certificateProperties/notValidAfter
Property: notValidAfter (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time according to ISO-8601 standard from which the certificate is not valid anymore

8.1.34.3.5 Algorithm Reference

Location: /components/[]/cryptoProperties/certificateProperties/signatureAlgorithmRef
Property: signatureAlgorithmRef (Optional)

Type: String

Description: The bom-ref to signature algorithm used by the certificate

8.1.34.3.6 Key reference

Location: /components/[]/cryptoProperties/certificateProperties/subjectPublicKeyRef
Property: subjectPublicKeyRef (Optional)

Type: String

Description: The bom-ref to the public key of the subject

8.1.34.3.7 Certificate Format

Location: /components/[]/cryptoProperties/certificateProperties/certificateFormat
Property: certificateFormat (Optional)

Type: String

Description: The format of the certificate

Examples:

- X.509
- PEM
- DER
- CVC

8.1.34.3.8 Certificate File Extension

Location: /components/[]/cryptoProperties/certificateProperties/certificateExtension
Property: certificateExtension (Optional)

Type: String

Description: The file extension of the certificate

Examples:

- crt
- pem
- cer
- der
- p12

8.1.34.4 Related Cryptographic Material Properties

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties
Property: relatedCryptoMaterialProperties (Optional)

Type: Object

Description: Properties for cryptographic assets of asset type: related-crypto-material

Table 183 – Properties for the relatedCryptoMaterialProperties object

Property	Type	Requirement	Description
type	String	Optional	The type for the related cryptographic material
id	String	Optional	The optional unique identifier for the related cryptographic material.
state	String	Optional	The key state as defined by NIST SP 800-57.
algorithmRef	String	Optional	The bom-ref to the algorithm used to generate the related cryptographic material.
creationDate	String	Optional	The date and time (timestamp) when the related cryptographic material was created.
activationDate	String	Optional	The date and time (timestamp) when the related cryptographic material was activated.
updateDate	String	Optional	The date and time (timestamp) when the related cryptographic material was updated.
expirationDate	String	Optional	The date and time (timestamp) when the related cryptographic material expires.
value	String	Optional	The associated value of the cryptographic material.
size	Integer	Optional	The size of the cryptographic asset (in bits).
format	String	Optional	The format of the related cryptographic material (e.g. P8, PEM, DER).
securedBy	Object	Optional	The mechanism by which the cryptographic asset is secured by.

8.1.34.4.1 relatedCryptoMaterialType

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/type
Property: type (Optional)

Type: String

Description: The type for the related cryptographic material

Table 184 – Enumeration of possible values

Value	Description
private-key	The confidential key of a key pair used in asymmetric cryptography.
public-key	The non-confidential key of a key pair used in asymmetric cryptography.
secret-key	A key used to encrypt and decrypt messages in symmetric cryptography.
key	A piece of information, usually an octet string, which, when processed through a cryptographic algorithm, processes cryptographic data.
ciphertext	The result of encryption performed on plaintext using an algorithm (or cipher).
signature	A cryptographic value that is calculated from the data and a key known only by the signer.
digest	The output of the hash function.
initialization-vector	A fixed-size random or pseudo-random value used as an input parameter for cryptographic algorithms.
nonce	A random or pseudo-random number that can only be used once in a cryptographic communication.
seed	The input to a pseudo-random number generator. Different seeds generate different pseudo-random sequences.
salt	A value used in a cryptographic process, usually to ensure that the results of computations for one instance cannot be reused by an attacker.
shared-secret	A piece of data known only to the parties involved, in a secure communication.
tag	A message authentication code (MAC), sometimes known as an authentication tag, is a short piece of information used for authenticating and integrity-checking a message.
additional-data	An unspecified collection of data with relevance to cryptographic activity.
password	A secret word, phrase, or sequence of characters used during authentication or authorization.
credential	Establishes the identity of a party to communication, usually in the form of cryptographic keys or passwords.
token	An object encapsulating a security identity.
other	Another type of cryptographic asset.
unknown	The type of cryptographic asset is not known.

8.1.34.4.2 ID

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/id

Property: id (Optional)

Type: String

Description: The optional unique identifier for the related cryptographic material.

8.1.34.4.3 State

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/state
Property: state (Optional)

Type: String

Description: The key state as defined by NIST SP 800-57.

Enumeration: Must be one of:

- pre-activation
- active
- suspended
- deactivated
- compromised
- destroyed

8.1.34.4.4 Algorithm Reference

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/algorithmRef
Property: algorithmRef (Optional)

Type: String

Description: The bom-ref to the algorithm used to generate the related cryptographic material.

8.1.34.4.5 Creation Date

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/creationDate
Property: creationDate (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time (timestamp) when the related cryptographic material was created.

8.1.34.4.6 Activation Date

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/activationDate
Property: activationDate (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time (timestamp) when the related cryptographic material was activated.

8.1.34.4.7 Update Date

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/updateDate
Property: updateDate (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time (timestamp) when the related cryptographic material was updated.

8.1.34.4.8 Expiration Date

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/expirationDate
Property: expirationDate (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time (timestamp) when the related cryptographic material expires.

8.1.34.4.9 Value

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/value
Property: value (Optional)

Type: String

Description: The associated value of the cryptographic material.

8.1.34.4.10 Size

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/size
Property: size (Optional)

Type: Integer

Description: The size of the cryptographic asset (in bits).

8.1.34.4.11 Format

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/format
Property: format (Optional)

Type: String

Description: The format of the related cryptographic material (e.g. P8, PEM, DER).

8.1.34.4.12 Secured By

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/securedBy
Property: securedBy (Optional)

Type: Object

Description: The mechanism by which the cryptographic asset is secured by.

Table 185 – Properties for the securedBy object

Property	Type	Requirement	Description
mechanism	String	Optional	Specifies the mechanism by which the cryptographic asset is secured by.
algorithmRef	String	Optional	The bom-ref to the algorithm.

8.1.34.4.13 Mechanism

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/securedBy/mechanism
Property: mechanism (Optional)

Type: String

Description: Specifies the mechanism by which the cryptographic asset is secured by.

Examples:

- HSM
- TPM
- SGX
- Software
- None

8.1.34.4.14 Algorithm Reference

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/securedBy/algorithmRef
Property: algorithmRef (Optional)

Type: String

Description: The bom-ref to the algorithm.

8.1.34.5 Protocol Properties

Location: /components/[]/cryptoProperties/protocolProperties
Property: protocolProperties (Optional)

Type: Object

Description: Properties specific to cryptographic assets of type: protocol.

Table 186 – Properties for the protocolProperties object

Property	Type	Requirement	Description
type	String	Optional	The concrete protocol type.
version	String	Optional	The version of the protocol.
cipherSuites	Array	Optional	A list of cipher suites related to the protocol.
ikev2TransformTypes	Object	Optional	The IKEv2 transform types supported (types 1-4), defined in RFC 7296 section 3.3.2, and additional properties.
cryptoRefArray	Array	Optional	A list of protocol-related cryptographic assets

8.1.34.5.1 Type

Location: /components/[]/cryptoProperties/protocolProperties/type
Property: type (Optional)

Type: String

Description: The concrete protocol type.

Table 187 – Enumeration of possible values

Value	Description
tls	Transport Layer Security
ssh	Secure Shell
ipsec	Internet Protocol Security
ike	Internet Key Exchange
sstp	Secure Socket Tunnelling Protocol
wpa	Wi-Fi Protected Access
other	Another protocol type
unknown	The protocol type is not known

8.1.34.5.2 Protocol Version

Location: /components/[]/cryptoProperties/protocolProperties/version
Property: version (Optional)

Type: String

Description: The version of the protocol.

Examples:

- 1.0
- 1.2
- 1.99

8.1.34.5.3 Cipher Suites

Location: /components/[]/cryptoProperties/protocolProperties/cipherSuites
Property: cipherSuites (Optional)

Type: Array

Description: A list of cipher suites related to the protocol. Each item of this array must be a Cipher Suite object.

8.1.34.5.4 Cipher Suite

Location: /components/[]/cryptoProperties/protocolProperties/cipherSuites/[]

Type: Object
Description: Object representing a cipher suite

Table 188 – Properties for the cipherSuites object

Property	Type	Requirement	Description
name	String	Optional	A common name for the cipher suite.
algorithms	Array	Optional	A list of algorithms related to the cipher suite.
identifiers	Array	Optional	A list of common identifiers for the cipher suite.

8.1.34.5.5 Common Name

Location: /components/[]/cryptoProperties/protocolProperties/cipherSuites/[]/name
Property: name (Optional)

Type: String

Description: A common name for the cipher suite.

Examples:

- TLS_DHE_RSA_WITH_AES_128_CCM

8.1.34.5.6 Related Algorithms

Location: /components/[]/cryptoProperties/protocolProperties/cipherSuites/[]/algorithms
Property: algorithms (Optional)

Type: array (of String)

Description: A list of algorithms related to the cipher suite. The bom-ref to algorithm cryptographic asset. Each item of this array must be a string.

8.1.34.5.7 Cipher Suite Identifiers

Location: /components/[]/cryptoProperties/protocolProperties/cipherSuites/[]/identifiers
Property: identifiers (Optional)

Type: array (of String)

Description: A list of common identifiers for the cipher suite. Cipher suite identifier Each item of this array must be a string.

Examples:

- 0xC0
- 0x9E

8.1.34.5.8 IKEv2 Transform Types

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes
Property: ikev2TransformTypes (Optional)

Type: Object

Description: The IKEv2 transform types supported (types 1-4), defined in RFC 7296 section 3.3.2, and additional properties.

Table 189 – Properties for the ikev2TransformTypes object

Property	Type	Requirement	Description
encr	Array	Optional	Transform Type 1: encryption algorithms
prf	Array	Optional	Transform Type 2: pseudorandom functions
integ	Array	Optional	Transform Type 3: integrity algorithms
ke	Array	Optional	Transform Type 4: Key Exchange Method (KE) per RFC 9370, formerly called Diffie-Hellman Group (D-H).
esn	Boolean	Optional	Specifies if an Extended Sequence Number (ESN) is used.
auth	Array	Optional	IKEv2 Authentication method

8.1.34.5.9 Encryption Algorithm (ENCR)

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/encr
Property: encr (Optional)

Type: array (of String)

Description: Transform Type 1: encryption algorithms Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links. Each item of this array must be a string.

8.1.34.5.10 Pseudorandom Function (PRF)

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/prf
Property: prf (Optional)

Type: array (of String)

Description: Transform Type 2: pseudorandom functions Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links. Each item of this array must be a string.

8.1.34.5.11 Integrity Algorithm (INTEG)

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/integ
Property: integ (Optional)

Type: array (of String)

Description: Transform Type 3: integrity algorithms Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links. Each item of this array must be a string.

8.1.34.5.12 Key Exchange Method (KE)

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/ke
Property: ke (Optional)

Type: array (of String)

Description: Transform Type 4: Key Exchange Method (KE) per RFC 9370, formerly called Diffie-Hellman Group (D-H). Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links. Each item of this array must be a string.

8.1.34.5.13 Extended Sequence Numbers (ESN)

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/esn
Property: esn (Optional)

Type: Boolean

Description: Specifies if an Extended Sequence Number (ESN) is used.

8.1.34.5.14 IKEv2 Authentication method

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/auth
Property: auth (Optional)

Type: array (of String)

Description: IKEv2 Authentication method Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links. Each item of this array must be a string.

8.1.34.5.15 Cryptographic References

Location: /components/[]/cryptoProperties/protocolProperties/cryptoRefArray
Property: cryptoRefArray (Optional)

Type: array (of String)

Description: A list of protocol-related cryptographic assets Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links. Each item of this array must be a string.

8.1.34.6 OID

Location: /components/[]/cryptoProperties/oid
Property: oid (Optional)

Type: String

Description: The object identifier (OID) of the cryptographic asset.

8.1.35 Properties

Location: /components/[]/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

8.1.35.1 Lightweight name-value pair

Location: /components/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 190 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

8.1.35.1.1 Name

Location: /components/[]/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

8.1.35.1.2 Value

Location: /components/[]/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

8.1.36 Tags

Location: /components/[]/tags
Property: tags (Optional)

Type: array (of String)

Description: Textual strings that aid in discovery, search, and retrieval of the associated object. Tags often serve as a way to group or categorize similar or related objects by various attributes. Each item of this array must be a string.

Examples:

- json-parser
- object-persistence
- text-to-image
- translation
- object-detection

8.1.37 Signature

Location: /components/[]/signature
Property: signature (Optional)

Type: Object

Description: An enveloped digital signature embedded within and specific to this object within the BOM. CycloneDX signatures enable integrity and authenticity verification without separating the signature from the BOM. Enveloped signatures enable each party in the supply chain to take responsibility for and sign their specific data, ensuring its integrity and authenticity. By aggregating all signatures, stakeholders can independently verify discrete pieces of information from each provider, enhancing overall transparency and trust in the supply chain.

Reference: Refer to the JSON Signature Format specification or to the XML Signature specification for implementation details.

- https://cyberphone.github.io/doc/security/jsf.html
- https://www.w3.org/TR/xmldsig-core/

All items must be unique.

9 Services

Location: /services
Property: services (Optional)

Type: Array

Description: A list of services. This may include microservices, function-as-a-service, and other types of network or intra-process services. Each item of this array must be a Service object.

9.1 Service

Location: /services/[]

Type: Object

Table 191 – Properties for the services object

Property	Type	Requirement	Description
bom-ref	String	Optional	Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
provider	Object	Optional	The organization that provides the service.
group	String	Optional	The grouping name, namespace, or identifier. This will often be a shortened, single name of the company or project that produced the service or domain name. Whitespace and special characters should be avoided.
name	String	Required	The name of the service. This will often be a shortened, single name of the service.
version	String	Optional	The service version.
description	String	Optional	Specifies a description for the service
endpoints	Array	Optional	The endpoint URIs of the service. Multiple endpoints are allowed.
authenticated	Boolean	Optional	A boolean value indicating if the service requires authentication. A value of true indicates the service requires authentication prior to use. A value of false indicates the service does not require authentication.
x-trust-boundary	Boolean	Optional	A boolean value indicating if use of the service crosses a trust zone or boundary. A value of true indicates that by using the service, a trust boundary is crossed. A value of false indicates that by using the service, a trust boundary is not crossed.
trustZone	String	Optional	The name of the trust zone the service resides in.
data	Array	Optional	Specifies information about the data including the directional flow of data and the data classification.

Property	Type	Requirement	Description
licenses	Array	Optional	EITHER (list of SPDX licenses and/or named licenses) OR (tuple of one SPDX License Expression)
externalReferences	Array	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.
services	Array	Optional	A list of services included or deployed behind the parent service. This is not a dependency tree. It provides a way to specify a hierarchical representation of service assemblies.
releaseNotes	Object	Optional	Specifies optional release notes.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.
tags	Array	Optional	Textual strings that aid in discovery, search, and retrieval of the associated object. Tags often serve as a way to group or categorize similar or related objects by various attributes.
signature	Array	Optional	Enveloped signature in JSON Signature Format (JSF).

9.1.1 BOM Reference

Location: /services/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

9.1.2 Provider

Location: /services/[]/provider
Property: provider (Optional)

Type: Object

Description: The organization that provides the service.

Table 192 – Properties for the provider object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

9.1.2.1 BOM Reference

Location: /services/[]/provider/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

9.1.2.2 Organization Name

Location: /services/[]/provider/name
Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

9.1.2.3 Organization Address

Location: /services/[]/provider/address
Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 193 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.

Property	Type	Requirement	Description
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

9.1.2.3.1 BOM Reference

Location: /services/[]/provider/address/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

9.1.2.3.2 Country

Location: /services/[]/provider/address/country
Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

9.1.2.3.3 Region

Location: /services/[]/provider/address/region
Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

9.1.2.3.4 Locality

Location: /services/[]/provider/address/locality
Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

9.1.2.3.5 Post Office Box Number

Location: /services/[]/provider/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

9.1.2.3.6 Postal Code

Location: /services/[]/provider/address/postalCode
Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

9.1.2.3.7 Street Address

Location: /services/[]/provider/address/streetAddress
Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

9.1.2.4 Organization URL(s)

Location: /services/[]/provider/url
Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

9.1.2.5 Organizational Contact

Location: /services/[]/provider/contact
Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

9.1.2.5.1 Organizational Contact

Location: /services/[]/provider/contact/[]

Type: Object

Table 194 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

9.1.2.5.2 BOM Reference

Location: /services/[]/provider/contact/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

9.1.2.5.3 Name

Location: /services/[]/provider/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

9.1.2.5.4 Email Address

Location: /services/[]/provider/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

9.1.2.5.5 Phone

Location: /services/[]/provider/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

9.1.3 Service Group

Location: /services/[]/group
Property: group (Optional)

Type: String

Description: The grouping name, namespace, or identifier. This will often be a shortened, single name of the company or project that produced the service or domain name. Whitespace and special characters should be avoided.

Examples:

- com.acme

9.1.4 Service Name

Location: /services/[]/name
Property: name (Required)

Type: String

Description: The name of the service. This will often be a shortened, single name of the service.

Examples:

- ticker-service

9.1.5 Service Version

Location: /services/[]/version
Property: version (Optional)

Type: String

Description: The service version.

Examples:

- 9.0.14
- v1.33.7
- 7.0.0-M1
- 2.0pre1
- 1.0.0-beta1
- 0.8.15

9.1.6 Service Description

Location: /services/[]/description
Property: description (Optional)

Type: String

Description: Specifies a description for the service

9.1.7 Endpoints

Location: /services/[]/endpoints
Property: endpoints (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The endpoint URIs of the service. Multiple endpoints are allowed. Each item of this array must be a string.

Examples:

- https://example.com/api/v1/ticker

9.1.8 Authentication Required

Location: /services/[]/authenticated
Property: authenticated (Optional)

Type: Boolean

Description: A boolean value indicating if the service requires authentication. A value of true indicates the service requires authentication prior to use. A value of false indicates the service does not require authentication.

9.1.9 Crosses Trust Boundary

Location: /services/[]/x-trust-boundary
Property: x-trust-boundary (Optional)

Type: Boolean

Description: A boolean value indicating if use of the service crosses a trust zone or boundary. A value of true indicates that by using the service, a trust boundary is crossed. A value of false indicates that by using the service, a trust boundary is not crossed.

9.1.10 Trust Zone

Location: /services/[]/trustZone
Property: trustZone (Optional)

Type: String

Description: The name of the trust zone the service resides in.

9.1.11 Data

Location: /services/[]/data
Property: data (Optional)

Type: Array

Description: Specifies information about the data including the directional flow of data and the data classification. Each item of this array must be a Hash Objects object.

9.1.11.1 Hash Objects

Location: /services/[]/data/[]

Type: Object

Table 195 – Properties for the data object

Property	Type	Requirement	Description
flow	String	Required	Specifies the flow direction of the data. Direction is relative to the service. Inbound flow states that data enters the service. Outbound flow states that data leaves the service. Bi-directional states that data flows both ways and unknown states that the direction is not known.
classification	String	Required	Data classification tags data according to its type, sensitivity, and value if altered, stolen, or destroyed.
name	String	Optional	Name for the defined data

Property	Type	Requirement	Description
description	String	Optional	Short description of the data content and usage
governance	Object	Optional	Data governance captures information regarding data ownership, stewardship, and custodianship, providing insights into the individuals or entities responsible for managing, overseeing, and safeguarding the data throughout its lifecycle.
source	Array	Optional	The URI, URL, or BOM-Link of the components or services the data came in from
destination	Array	Optional	The URI, URL, or BOM-Link of the components or services the data is sent to

9.1.11.1.1 Directional Flow

Location: /services/[]/data/[]/flow
Property: flow (Required)

Type: String

Description: Specifies the flow direction of the data. Direction is relative to the service. Inbound flow states that data enters the service. Outbound flow states that data leaves the service. Bi-directional states that data flows both ways and unknown states that the direction is not known.

Table 196 – Enumeration of possible values

Value	Description
inbound	Data that enters a service.
outbound	Data that exits a service.
bi-directional	Data flows in and out of the service.
unknown	The directional flow of data is not known.

9.1.11.1.2 Data Classification

Location: /services/[]/data/[]/classification
Property: classification (Required)

Type: String

Description: Data classification tags data according to its type, sensitivity, and value if altered, stolen, or destroyed.

9.1.11.1.3 Name

Location: /services/[]/data/[]/name
Property: name (Optional)

Type: String

Description: Name for the defined data

Examples:

- Credit card reporting

9.1.11.1.4 Description

Location: /services/[]/data/[]/description
Property: description (Optional)

Type: String

Description: Short description of the data content and usage

Examples:

- Credit card information being exchanged in between the web app and the database

9.1.11.1.5 Data Governance

Location: /services/[]/data/[]/governance
Property: governance (Optional)

Type: Object

Description: Data governance captures information regarding data ownership, stewardship, and custodianship, providing insights into the individuals or entities responsible for managing, overseeing, and safeguarding the data throughout its lifecycle.

Table 197 – Properties for the governance object

Property	Type	Requirement	Description
custodians	Array	Optional	Data custodians are responsible for the safe custody, transport, and storage of data.
stewards	Array	Optional	Data stewards are responsible for data content, context, and associated business rules.
owners	Array	Optional	Data owners are concerned with risk and appropriate access to data.

9.1.11.1.6 Data Custodians

Location: /services/[]/data/[]/governance/custodians
Property: custodians (Optional)

Type: Array

Description: Data custodians are responsible for the safe custody, transport, and storage of data.

Location: /services/[]/data/[]/governance/custodians/[]

Type: Object

Table 198 – Properties for the custodians object

Property	Type	Requirement	Description
organization	Object	Optional	The organization that is responsible for specific data governance role(s).
contact	Object	Optional	The individual that is responsible for specific data governance role(s).

9.1.11.1.7 Organization

Location: /services/[]/data/[]/governance/custodians/[]/organization
Property: organization (Optional)

Type: Object

Description: The organization that is responsible for specific data governance role(s).

Table 199 – Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

9.1.11.1.8 BOM Reference

Location: /services/[]/data/[]/governance/custodians/[]/organization/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

9.1.11.1.9 Organization Name

Location: /services/[]/data/[]/governance/custodians/[]/organization/name
Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

9.1.11.1.10 Organization Address

Location: /services/[]/data/[]/governance/custodians/[]/organization/address
Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 200 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

9.1.11.1.11 BOM Reference

Location: /services/[]/data/[]/governance/custodians/[]/organization/address/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

9.1.11.1.12 Country

Location: /services/[]/data/[]/governance/custodians/[]/organization/address/country
Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

9.1.11.1.13 Region

Location: /services/[]/data/[]/governance/custodians/[]/organization/address/region
Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

9.1.11.1.14 Locality

Location: /services/[]/data/[]/governance/custodians/[]/organization/address/locality
Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

9.1.11.1.15 Post Office Box Number

Location: /services/[]/data/[]/governance/custodians/[]/organization/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

9.1.11.1.16 Postal Code

Location: /services/[]/data/[]/governance/custodians/[]/organization/address/postalCode
Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

9.1.11.1.17 Street Address

Location: /services/[]/data/[]/governance/custodians/[]/organization/address/streetAddress
Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

9.1.11.1.18 Organization URL(s)

Location: /services/[]/data/[]/governance/custodians/[]/organization/url
Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

9.1.11.1.19 Organizational Contact

Location: /services/[]/data/[]/governance/custodians/[]/organization/contact
Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

9.1.11.1.20 Organizational Contact

Location: /services/[]/data/[]/governance/custodians/[]/organization/contact/[]

Type: Object

Table 201 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

9.1.11.1.21 BOM Reference

Location: /services/[]/data/[]/governance/custodians/[]/organization/contact/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

9.1.11.1.22 Name

Location: /services/[]/data/[]/governance/custodians/[]/organization/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

9.1.11.1.23 Email Address

Location: /services/[]/data/[]/governance/custodians/[]/organization/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

9.1.11.1.24 Phone

Location: /services/[]/data/[]/governance/custodians/[]/organization/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

9.1.11.1.25 Organizational Contact

Location: /services/[]/data/[]/governance/custodians/[]/contact
Property: contact (Optional)

Type: Object

Description: The individual that is responsible for specific data governance role(s).

Table 202 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

9.1.11.1.26 BOM Reference

Location: /services/[]/data/[]/governance/custodians/[]/contact/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

9.1.11.1.27 Name

Location: /services/[]/data/[]/governance/custodians/[]/contact/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

9.1.11.1.28 Email Address

Location: /services/[]/data/[]/governance/custodians/[]/contact/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

9.1.11.1.29 Phone

Location: /services/[]/data/[]/governance/custodians/[]/contact/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

9.1.11.1.30 Data Stewards

Location: /services/[]/data/[]/governance/stewards
Property: stewards (Optional)

Type: Array

Description: Data stewards are responsible for data content, context, and associated business rules.

Location: /services/[]/data/[]/governance/stewards/[]

Type: Object

Table 203 – Properties for the stewards object

Property	Type	Requirement	Description
organization	Object	Optional	The organization that is responsible for specific data governance role(s).
contact	Object	Optional	The individual that is responsible for specific data governance role(s).

9.1.11.1.31 Organization

Location: /services/[]/data/[]/governance/stewards/[]/organization
Property: organization (Optional)

Type: Object

Description: The organization that is responsible for specific data governance role(s).

Table 204 – Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

9.1.11.1.32 BOM Reference

Location: /services/[]/data/[]/governance/stewards/[]/organization/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

9.1.11.1.33 Organization Name

Location: /services/[]/data/[]/governance/stewards/[]/organization/name
Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

9.1.11.1.34 Organization Address

Location: /services/[]/data/[]/governance/stewards/[]/organization/address
Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 205 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

9.1.11.1.35 BOM Reference

Location: /services/[]/data/[]/governance/stewards/[]/organization/address/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

9.1.11.1.36 Country

Location: /services/[]/data/[]/governance/stewards/[]/organization/address/country
Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

9.1.11.1.37 Region

Location: /services/[]/data/[]/governance/stewards/[]/organization/address/region
Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

9.1.11.1.38 Locality

Location: /services/[]/data/[]/governance/stewards/[]/organization/address/locality
Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

9.1.11.1.39 Post Office Box Number

Location: /services/[]/data/[]/governance/stewards/[]/organization/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

9.1.11.1.40 Postal Code

Location: /services/[]/data/[]/governance/stewards/[]/organization/address/postalCode
Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

9.1.11.1.41 Street Address

Location: /services/[]/data/[]/governance/stewards/[]/organization/address/streetAddress
Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

9.1.11.1.42 Organization URL(s)

Location: /services/[]/data/[]/governance/stewards/[]/organization/url
Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

9.1.11.1.43 Organizational Contact

Location: /services/[]/data/[]/governance/stewards/[]/organization/contact
Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

9.1.11.1.44 Organizational Contact

Location: /services/[]/data/[]/governance/stewards/[]/organization/contact/[]

Type: Object

Table 206 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

9.1.11.1.45 BOM Reference

Location: /services/[]/data/[]/governance/stewards/[]/organization/contact/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

9.1.11.1.46 Name

Location: /services/[]/data/[]/governance/stewards/[]/organization/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

9.1.11.1.47 Email Address

Location: /services/[]/data/[]/governance/stewards/[]/organization/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

9.1.11.1.48 Phone

Location: /services/[]/data/[]/governance/stewards/[]/organization/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

9.1.11.1.49 Organizational Contact

Location: /services/[]/data/[]/governance/stewards/[]/contact
Property: contact (Optional)

Type: Object

Description: The individual that is responsible for specific data governance role(s).

Table 207 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

9.1.11.1.50 BOM Reference

Location: /services/[]/data/[]/governance/stewards/[]/contact/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

9.1.11.1.51 Name

Location: /services/[]/data/[]/governance/stewards/[]/contact/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

9.1.11.1.52 Email Address

Location: /services/[]/data/[]/governance/stewards/[]/contact/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

9.1.11.1.53 Phone

Location: /services/[]/data/[]/governance/stewards/[]/contact/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

9.1.11.1.54 Data Owners

Location: /services/[]/data/[]/governance/owners
Property: owners (Optional)

Type: Array

Description: Data owners are concerned with risk and appropriate access to data.

Location: /services/[]/data/[]/governance/owners/[]

Type: Object

Table 208 – Properties for the owners object

Property	Type	Requirement	Description
organization	Object	Optional	The organization that is responsible for specific data governance role(s).
contact	Object	Optional	The individual that is responsible for specific data governance role(s).

9.1.11.1.55 Organization

Location: /services/[]/data/[]/governance/owners/[]/organization
Property: organization (Optional)

Type: Object

Description: The organization that is responsible for specific data governance role(s).

Table 209 – Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

9.1.11.1.56 BOM Reference

Location: /services/[]/data/[]/governance/owners/[]/organization/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

9.1.11.1.57 Organization Name

Location: /services/[]/data/[]/governance/owners/[]/organization/name
Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

9.1.11.1.58 Organization Address

Location: /services/[]/data/[]/governance/owners/[]/organization/address

Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 210 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

9.1.11.1.59 BOM Reference

Location: /services/[]/data/[]/governance/owners/[]/organization/address/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

9.1.11.1.60 Country

Location: /services/[]/data/[]/governance/owners/[]/organization/address/country
Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

9.1.11.1.61 Region

Location: /services/[]/data/[]/governance/owners/[]/organization/address/region
Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

9.1.11.1.62 Locality

Location: /services/[]/data/[]/governance/owners/[]/organization/address/locality
Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

9.1.11.1.63 Post Office Box Number

Location: /services/[]/data/[]/governance/owners/[]/organization/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

9.1.11.1.64 Postal Code

Location: /services/[]/data/[]/governance/owners/[]/organization/address/postalCode
Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

9.1.11.1.65 Street Address

Location: /services/[]/data/[]/governance/owners/[]/organization/address/streetAddress
Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

9.1.11.1.66 Organization URL(s)

Location: /services/[]/data/[]/governance/owners/[]/organization/url
Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

9.1.11.1.67 Organizational Contact

Location: /services/[]/data/[]/governance/owners/[]/organization/contact

Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

9.1.11.1.68 Organizational Contact

Location: /services/[]/data/[]/governance/owners/[]/organization/contact/[]

Type: Object

Table 211 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

9.1.11.1.69 BOM Reference

Location: /services/[]/data/[]/governance/owners/[]/organization/contact/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

9.1.11.1.70 Name

Location: /services/[]/data/[]/governance/owners/[]/organization/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

9.1.11.1.71 Email Address

Location: /services/[]/data/[]/governance/owners/[]/organization/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

9.1.11.1.72 Phone

Location: /services/[]/data/[]/governance/owners/[]/organization/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

9.1.11.1.73 Organizational Contact

Location: /services/[]/data/[]/governance/owners/[]/contact
Property: contact (Optional)

Type: Object

Description: The individual that is responsible for specific data governance role(s).

Table 212 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

9.1.11.1.74 BOM Reference

Location: /services/[]/data/[]/governance/owners/[]/contact/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

9.1.11.1.75 Name

Location: /services/[]/data/[]/governance/owners/[]/contact/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

9.1.11.1.76 Email Address

Location: /services/[]/data/[]/governance/owners/[]/contact/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

9.1.11.1.77 Phone

Location: /services/[]/data/[]/governance/owners/[]/contact/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

9.1.11.1.78 Source

Location: /services/[]/data/[]/source
Property: source (Optional)

Type: Array

Description: The URI, URL, or BOM-Link of the components or services the data came in from

Location: /services/[]/data/[]/source/[]

*Must be **any of**:*

1. URL
2. BOM-Link Element

9.1.11.1.79 URL

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

9.1.11.1.80 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

9.1.11.1.81 Destination

Location: /services/[]/data/[]/destination
Property: destination (Optional)

Type: Array

Description: The URI, URL, or BOM-Link of the components or services the data is sent to

Location: /services/[]/data/[]/destination/[]

*Must be **any of**:*

1. URL
2. BOM-Link Element

9.1.11.1.82 URL

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

9.1.11.1.83 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$

Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

9.1.12 Service Licence(s)

Location: /services/[]/licenses

Property: licenses (Optional)

Type: Array

Description: EITHER (list of SPDX licences and/or named licences) OR (tuple of one SPDX Licence Expression)

Reference: Refer to the license definition at /components/[]/licenses

9.1.13 External References

Location: /services/[]/externalReferences

Property: externalReferences (Optional)

Type: Array

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM. Each item of this array must be an External Reference object.

Reference: Refer to the external reference definition at /externalReferences/[]

9.1.13.1 External Reference

Location: /services/[]/externalReferences/[]

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

9.1.14 Services

Location: /services/[]/services

Property: services (Optional)

Type: Array

Description: A list of services included or deployed behind the parent service. This is not a dependency tree. It provides a way to specify a hierarchical representation of service assemblies. Each item of this array must be a Service object.

9.1.14.1 Service

Location: /services/[]/services/[]

Type: Object

Reference: Refer to the service definition at /services/[]

All items must be unique.

9.1.15 Release notes

Location: /services/[]/releaseNotes
Property: releaseNotes (Optional)

Type: Object

Description: Specifies optional release notes.

Table 213 – Properties for the releaseNotes object

Property	Type	Requirement	Description
type	String	Required	The software versioning type the release note describes.
title	String	Optional	The title of the release.
featuredImage	String	Optional	The URL to an image that may be prominently displayed with the release note.
socialImage	String	Optional	The URL to an image that may be used in messaging on social media platforms.
description	String	Optional	A short description of the release.
timestamp	String	Optional	The date and time (timestamp) when the release note was created.
aliases	Array	Optional	One or more alternate names the release may be referred to. This may include unofficial terms used by development and marketing teams (e.g. code names).
tags	Array	Optional	Textual strings that aid in discovery, search, and retrieval of the associated object. Tags often serve as a way to group or categorize similar or related objects by various attributes.
resolves	Array	Optional	A collection of issues that have been resolved.
notes	Array	Optional	Zero or more release notes containing the locale and content. Multiple note objects may be specified to support release notes in a wide variety of languages.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

9.1.15.1 Type

Location: /services/[]/releaseNotes/type
Property: type (Required)

Type: String

Description: The software versioning type the release note describes.

Examples:

- major
- minor
- patch
- pre-release
- internal

9.1.15.2 Title

Location: /services/[]/releaseNotes/title
Property: title (Optional)

Type: String

Description: The title of the release.

9.1.15.3 Featured image

Location: /services/[]/releaseNotes/featuredImage
Property: featuredImage (Optional)

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL to an image that may be prominently displayed with the release note.

9.1.15.4 Social image

Location: /services/[]/releaseNotes/socialImage
Property: socialImage (Optional)

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL to an image that may be used in messaging on social media platforms.

9.1.15.5 Description

Location: /services/[]/releaseNotes/description
Property: description (Optional)

Type: String

Description: A short description of the release.

9.1.15.6 Timestamp

Location: /services/[]/releaseNotes/timestamp
Property: timestamp (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time (timestamp) when the release note was created.

9.1.15.7 Aliases

Location: /services/[]/releaseNotes/aliases
Property: aliases (Optional)

Type: array (of String)

Description: One or more alternate names the release may be referred to. This may include unofficial terms used by development and marketing teams (e.g. code names). Each item of this array must be a string.

9.1.15.8 Tags

Location: /services/[]/releaseNotes/tags
Property: tags (Optional)

Type: array (of String)

Description: Textual strings that aid in discovery, search, and retrieval of the associated object. Tags often serve as a way to group or categorize similar or related objects by various attributes. Each item of this array must be a string.

Examples:

- json-parser
- object-persistence
- text-to-image
- translation
- object-detection

9.1.15.9 Resolves

Location: /services/[]/releaseNotes/resolves
Property: resolves (Optional)

Type: Array

Description: A collection of issues that have been resolved. Each item of this array must be an Issue object.

9.1.15.9.1 Issue

Location: /services/[]/releaseNotes/resolves/[]

Type: Object
Description: An individual issue that has been resolved.

Table 214 – Properties for the resolves object

Property	Type	Requirement	Description
type	String	Required	Specifies the type of issue
id	String	Optional	The identifier of the issue assigned by the source of the issue
name	String	Optional	The name of the issue
description	String	Optional	A description of the issue
source	Object	Optional	The source of the issue where it is documented
references	Array	Optional	A collection of URL's for reference. Multiple URLs are allowed.

9.1.15.9.2 Issue Type

Location: /services/[]/releaseNotes/resolves/[]/type
Property: type (Required)

Type: String

Description: Specifies the type of issue

Table 215 – Enumeration of possible values

Value	Description
defect	A fault, flaw, or bug in software.
enhancement	A new feature or behaviour in software.
security	A special type of defect which impacts security.

9.1.15.9.3 Issue ID

Location: /services/[]/releaseNotes/resolves/[]/id
Property: id (Optional)

Type: String

Description: The identifier of the issue assigned by the source of the issue

9.1.15.9.4 Issue Name

Location: /services/[]/releaseNotes/resolves/[]/name
Property: name (Optional)

Type: String

Description: The name of the issue

9.1.15.9.5 Issue Description

Location: /services/[]/releaseNotes/resolves/[]/description
Property: description (Optional)

Type: String

Description: A description of the issue

9.1.15.9.6 Source

Location: /services/[]/releaseNotes/resolves/[]/source
Property: source (Optional)

Type: Object

Description: The source of the issue where it is documented

Table 216 – Properties for the source object

Property	Type	Requirement	Description
name	String	Optional	The name of the source.
url	String	Optional	The url of the issue documentation as provided by the source

9.1.15.9.7 Name

Location: /services/[]/releaseNotes/resolves/[]/source/name

Property: name (Optional)

Type: String

Description: The name of the source.

Examples:

- National Vulnerability Database
- NVD
- Apache

9.1.15.9.8 URL

Location: /services/[]/releaseNotes/resolves/[]/source/url
Property: url (Optional)

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The url of the issue documentation as provided by the source

9.1.15.9.9 References

Location: /services/[]/releaseNotes/resolves/[]/references
Property: references (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: A collection of URL's for reference. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

9.1.15.10 Notes

Location: /services/[]/releaseNotes/notes
Property: notes (Optional)

Type: Array

Description: Zero or more release notes containing the locale and content. Multiple note objects may be specified to support release notes in a wide variety of languages. Each item of this array must be a Note object.

9.1.15.10.1 Note

Location: /services/[]/releaseNotes/notes/[]

Type: Object
Description: A note containing the locale and content.

Table 217 – Properties for the notes object

Property	Type	Requirement	Description
locale	String	Optional	The ISO-639 (or higher) language code and optional ISO-3166 (or higher) country code. Examples include: "en", "en-US", "fr" and "fr-CA"
text	Object	Required	Specifies the full content of the release note.

9.1.15.10.2 Locale

Location: /services/[]/releaseNotes/notes/[]/locale
Property: locale (Optional)

Type: String

Pattern Constraint: ^([a-z]{2})(-[A-Z]{2})?$

Description: The ISO-639 (or higher) language code and optional ISO-3166 (or higher) country code. Examples include: "en", "en-US", "fr" and "fr-CA"

9.1.15.10.3 Release note content

Location: /services/[]/releaseNotes/notes/[]/text
Property: text (Required)

Type: Object

Description: Specifies the full content of the release note.

Table 218 – Properties for the text object

Property	Type	Requirement	Description
contentType	String	Optional	Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.
encoding	String	Optional	Specifies the optional encoding the text is represented in.
content	String	Required	The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

9.1.15.10.4 Content-Type

Location: /services/[]/releaseNotes/notes/[]/text/contentType
Property: contentType (Optional)

Type: String

Default Value: text/plain

Description: Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.

Examples:

- text/plain
- application/json
- image/png

9.1.15.10.5 Encoding

Location: /services/[]/releaseNotes/notes/[]/text/encoding
Property: encoding (Optional)

Type: String

Description: Specifies the optional encoding the text is represented in.

Table 219 – Enumeration of possible values

Value	Description
base64	Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string.

9.1.15.10.6 Attachment Text

Location: /services/[]/releaseNotes/notes/[]/text/content
Property: content (Required)

Type: String

Description: The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

9.1.15.11 Properties

Location: /services/[]/releaseNotes/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

9.1.15.11.1 Lightweight name-value pair

Location: /services/[]/releaseNotes/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 220 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

9.1.15.11.2 Name

Location: /services/[]/releaseNotes/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

9.1.15.11.3 Value

Location: /services/[]/releaseNotes/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

9.1.16 Properties

Location: /services/[]/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

9.1.16.1 Lightweight name-value pair

Location: /services/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 221 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

9.1.16.1.1 Name

Location: /services/[]/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

9.1.16.1.2 Value

Location: /services/[]/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

9.1.17 Tags

Location: /services/[]/tags
Property: tags (Optional)

Type: array (of String)

Description: Textual strings that aid in discovery, search, and retrieval of the associated object. Tags often serve as a way to group or categorize similar or related objects by various attributes. Each item of this array must be a string.

Examples:

- json-parser
- object-persistence
- text-to-image
- translation
- object-detection

9.1.18 Signature

Location: /services/[]/signature
Property: signature (Optional)

Type: Object

Description: An enveloped digital signature embedded within and specific to this object within the BOM. CycloneDX signatures enable integrity and authenticity verification without separating the signature from the BOM. Enveloped signatures enable each party in the supply chain to take responsibility for and sign their specific data, ensuring its integrity and authenticity. By aggregating all signatures, stakeholders can independently verify discrete pieces of information from each provider, enhancing overall transparency and trust in the supply chain.

Reference: Refer to the JSON Signature Format specification or to the XML Signature specification for implementation details.

- https://cyberphone.github.io/doc/security/jsf.html
- https://www.w3.org/TR/xmldsig-core/

All items must be unique.

10 External References

Location: /externalReferences
Property: externalReferences (Optional)

Type: Array

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM. Each item of this array must be an External Reference object.

10.1 External Reference

Location: /externalReferences/[]

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Table 222 – Properties for the externalReferences object

Property	Type	Requirement	Description
url	Array	Required	The URI (URL or URN) to the external reference. External references are URIs and therefore can accept any URL scheme including https (RFC-7230), mailto (RFC-2368), tel (RFC-3966), and dns (RFC-4501). External references may also include formally registered URNs such as CycloneDX BOM-Link to reference CycloneDX BOMs or any object within a BOM. BOM-Link transforms applicable external references into relationships that can be expressed in a BOM or across BOMs.
comment	String	Optional	An optional comment describing the external reference
type	String	Required	Specifies the type of external reference.
hashes	Array	Optional	The hashes of the external reference (if applicable).

10.1.1 URL

Location: /externalReferences/[]/url
Property: url (Required)

Description: The URI (URL or URN) to the external reference. External references are URIs and therefore can accept any URL scheme including https (RFC-7230), mailto (RFC-2368), tel (RFC-3966), and dns (RFC-4501). External references may also include formally registered URNs such as CycloneDX BOM-Link to reference CycloneDX BOMs or any object within a BOM. BOM-Link transforms applicable external references into relationships that can be expressed in a BOM or across BOMs.

*Must be **any of**:*

1. URL

10.1.2 URL

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

10.1.3 BOM-Link

10.1.4 Comment

Location: /externalReferences/[]/comment
Property: comment (Optional)

Type: String

Description: An optional comment describing the external reference

10.1.5 Type

Location: /externalReferences/[]/type
Property: type (Required)

Type: String

Description: Specifies the type of external reference.

Table 223 – Enumeration of possible values

Value	Description
vcs	Version Control System
issue-tracker	Issue or defect tracking system, or an Application Lifecycle Management (ALM) system
website	Website
advisories	Security advisories
bom	Bill of Materials (SBOM, OBOM, HBOM, SaaSBOM, etc)
mailing-list	Mailing list or discussion group
social	Social media account
chat	Real-time chat platform
documentation	Documentation, guides, or how-to instructions
support	Community or commercial support
source-distribution	The location where the source code distributable can be obtained. This is often an archive format such as zip or tgz. The source-distribution type complements use of the version control (vcs) type.
distribution	Direct or repository download location
distribution-intake	The location where a component was published to. This is often the same as "distribution" but may also include specialized publishing processes that act as an intermediary.
license	The reference to the licence file. If a licence URL has been defined in the licence node, it should also be defined as an external reference for completeness.
build-meta	Build-system specific meta file (i.e. pom.xml, package.json, .nuspec, etc)
build-system	Reference to an automated build system
release-notes	Reference to release notes
security-contact	Specifies a way to contact the maintainer, supplier, or provider in the event of a security incident. Common URIs include links to a disclosure procedure, a mailto (RFC-2368) that specifies an email address, a tel (RFC-3966) that specifies a phone number, or dns (RFC-4501) that specifies the records containing DNS Security TXT.
model-card	A model card describes the intended uses of a machine learning model, potential limitations, biases, ethical considerations, training parameters, datasets used to train the model, performance metrics, and other relevant data useful for ML transparency.
log	A record of events that occurred in a computer system or application, such as problems, errors, or information on current operations.
configuration	Parameters or settings that may be used by other components or services.

Value	Description
evidence	Information used to substantiate a claim.
formulation	Describes how a component or service was manufactured or deployed.
attestation	Human or machine-readable statements containing facts, evidence, or testimony.
threat-model	An enumeration of identified weaknesses, threats, and countermeasures, dataflow diagram (DFD), attack tree, and other supporting documentation in human-readable or machine-readable format.
adversary-model	The defined assumptions, goals, and capabilities of an adversary.
risk-assessment	Identifies and analyses the potential of future events that may negatively impact individuals, assets, and/or the environment. Risk assessments may also include judgments on the tolerability of each risk.
vulnerability-assertion	A Vulnerability Disclosure Report (VDR) which asserts the known and previously unknown vulnerabilities that affect a component, service, or product including the analysis and findings describing the impact (or lack of impact) that the reported vulnerability has on a component, service, or product.
exploitability-statement	A Vulnerability Exploitability eXchange (VEX) which asserts the known vulnerabilities that do not affect a product, product family, or organization, and optionally the ones that do. The VEX should include the analysis and findings describing the impact (or lack of impact) that the reported vulnerability has on the product, product family, or organization.
pentest-report	Results from an authorized simulated cyberattack on a component or service, otherwise known as a penetration test.
static-analysis-report	SARIF or proprietary machine or human-readable report for which static analysis has identified code quality, security, and other potential issues with the source code.
dynamic-analysis-report	Dynamic analysis report that has identified issues such as vulnerabilities and misconfigurations.
runtime-analysis-report	Report generated by analyzing the call stack of a running application.
component-analysis-report	Report generated by Software Composition Analysis (SCA), container analysis, or other forms of component analysis.
maturity-report	Report containing a formal assessment of an organization, business unit, or team against a maturity model.
certification-report	Industry, regulatory, or other certification from an accredited (if applicable) certification body.
codified-infrastructure	Code or configuration that defines and provisions virtualized infrastructure, commonly referred to as Infrastructure as Code (IaC).
quality-metrics	Report or system in which quality metrics can be obtained.

Value	Description
poam	Plans of Action and Milestones (POA&M) complement an "attestation" external reference. POA&M is defined by NIST as a "document that identifies tasks needing to be accomplished. It details resources required to accomplish the elements of the plan, any milestones in meeting the tasks and scheduled completion dates for the milestones".
electronic-signature	An e-signature is commonly a scanned representation of a written signature or a stylized script of the person's name.
digital-signature	A signature that leverages cryptography, typically public/private key pairs, which provides strong authenticity verification.
rfc-9116	Document that complies with RFC 9116 (A File Format to Aid in Security Vulnerability Disclosure)
other	Use this if no other types accurately describe the purpose of the external reference.

10.1.6 Hashes

Location: /externalReferences/[]/hashes
Property: hashes (Optional)

Type: Array

Description: The hashes of the external reference (if applicable). Each item of this array must be a Hash object.

10.1.6.1 Hash

Location: /externalReferences/[]/hashes/[]

Type: Object

Table 224 – Properties for the hashes object

Property	Type	Requirement	Description
alg	String	Required	The algorithm that generated the hash value.
content	String	Required	The value of the hash.

10.1.6.1.1 Hash Algorithm

Location: /externalReferences/[]/hashes/[]/alg
Property: alg (Required)

Type: String

Description: The algorithm that generated the hash value.

Enumeration: Must be one of:

- MD5
- SHA-1
- SHA-256
- SHA-384
- SHA-512

- SHA3-256
- SHA3-384
- SHA3-512
- BLAKE2b-256
- BLAKE2b-384
- BLAKE2b-512
- BLAKE3

10.1.6.1.2 Hash Value

Location: /externalReferences/[]/hashes/[]/content
Property: content (Required)

Type: String

Pattern Constraint: ^([a-fA-F0-9]{32}|[a-fA-F0-9]{40}|[a-fA-F0-9]{64}|[a-fA-F0-9]{96}|[a-fA-F0-9]{128})$

Description: The value of the hash.

Examples:

- 3942447fac867ae5cdb3229b658f4d48

11 Dependencies

Location: /dependencies
Property: dependencies (Optional)

Type: Array

Description: Provides the ability to document dependency relationships including provided & implemented components. Each item of this array must be a Dependency object.

11.1 Dependency

Location: /dependencies/[]

Type: Object

Description: Defines the direct dependencies of a component, service, or the components provided/implemented by a given component. Components or services that do not have their own dependencies must be declared as empty elements within the graph. Components or services that are not represented in the dependency graph may have unknown dependencies. It is recommended that implementations assume this to be opaque and not an indicator of an object being dependency-free. It is recommended to leverage compositions to indicate unknown dependency graphs.

Table 225 – Properties for the dependencies object

Property	Type	Requirement	Description
ref	String	Required	References a component or service by its bom-ref attribute
dependsOn	Array	Optional	The bom-ref identifiers of the components or services that are dependencies of this dependency object.
provides	Array	Optional	The bom-ref identifiers of the components or services that define a given specification or standard, which are provided or implemented by this dependency object. For example, a cryptographic library which implements a

Property	Type	Requirement	Description
			cryptographic algorithm. A component which implements another component does not imply that the implementation is in use.

11.1.1 Reference

Location: /dependencies/[]/ref
Property: ref (Required)

Type: String

Description: References a component or service by its bom-ref attribute

11.1.2 Depends On

Location: /dependencies/[]/dependsOn
Property: dependsOn (Optional)

Type: array (of String)

Description: The bom-ref identifiers of the components or services that are dependencies of this dependency object. Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType. Each item of this array must be a string.

All items must be unique.

11.1.3 Provides

Location: /dependencies/[]/provides
Property: provides (Optional)

Type: array (of String)

Description: The bom-ref identifiers of the components or services that define a given specification or standard, which are provided or implemented by this dependency object. For example, a cryptographic library which implements a cryptographic algorithm. A component which implements another component does not imply that the implementation is in use. Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType. Each item of this array must be a string.

All items must be unique.

12 Compositions

Location: /compositions
Property: compositions (Optional)

Type: Array

Description: Compositions describe constituent parts (including components, services, and dependency relationships) and their completeness. The completeness of vulnerabilities expressed in a BOM may also be described. Each item of this array must be a Compositions object.

12.1 Compositions

Location: /compositions/[]

Type: Object

Table 226 – Properties for the compositions object

Property	Type	Requirement	Description
bom-ref	String	Optional	Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
aggregate	String	Required	Specifies an aggregate type that describe how complete a relationship is.
assemblies	Array	Optional	The bom-ref identifiers of the components or services being described. Assemblies refer to nested relationships whereby a constituent part may include other constituent parts. References do not cascade to child parts. References are explicit for the specified constituent part only.
dependencies	Array	Optional	The bom-ref identifiers of the components or services being described. Dependencies refer to a relationship whereby an independent constituent part requires another independent constituent part. References do not cascade to transitive dependencies. References are explicit for the specified dependency only.
vulnerabilities	Array	Optional	The bom-ref identifiers of the vulnerabilities being described.
signature	Array	Optional	Enveloped signature in JSON Signature Format (JSF).

12.1.1 BOM Reference

Location: /compositions/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

12.1.2 Aggregate

Location: /compositions/[]/aggregate
Property: aggregate (Required)

Type: String

Default Value: not_specified

Description: Specifies an aggregate type that describe how complete a relationship is.

Table 227 – Enumeration of possible values

Value	Description
complete	The relationship is complete. No further relationships including constituent components, services, or dependencies are known to exist.
incomplete	The relationship is incomplete. Additional relationships exist and may include constituent components, services, or dependencies.
incomplete_first_party_only	The relationship is incomplete. Only relationships for first-party components, services, or their dependencies are represented.

Value	Description
incomplete_first_party_proprietary_only	The relationship is incomplete. Only relationships for first-party components, services, or their dependencies are represented, limited specifically to those that are proprietary.
incomplete_first_party_opensource_only	The relationship is incomplete. Only relationships for first-party components, services, or their dependencies are represented, limited specifically to those that are opensource.
incomplete_third_party_only	The relationship is incomplete. Only relationships for third-party components, services, or their dependencies are represented.
incomplete_third_party_proprietary_only	The relationship is incomplete. Only relationships for third-party components, services, or their dependencies are represented, limited specifically to those that are proprietary.
incomplete_third_party_opensource_only	The relationship is incomplete. Only relationships for third-party components, services, or their dependencies are represented, limited specifically to those that are opensource.
unknown	The relationship may be complete or incomplete. This usually signifies a 'best-effort' to obtain constituent components, services, or dependencies but the completeness is inconclusive.
not_specified	The relationship completeness is not specified.

12.1.3 BOM references

Location: /compositions/[]/assemblies
Property: assemblies (Optional)

Type: Array

Description: The bom-ref identifiers of the components or services being described. Assemblies refer to nested relationships whereby a constituent part may include other constituent parts. References do not cascade to child parts. References are explicit for the specified constituent part only.

Location: /compositions/[]/assemblies/[]

*Must be **any of**:*

1. Ref
2. BOM-Link Element

12.1.3.1 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

12.1.3.2 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

All items must be unique.

12.1.4 BOM references

Location: /compositions/[]/dependencies
Property: dependencies (Optional)

Type: array (of String)

Description: The bom-ref identifiers of the components or services being described. Dependencies refer to a relationship whereby an independent constituent part requires another independent constituent part. References do not cascade to transitive dependencies. References are explicit for the specified dependency only. Each item of this array must be a string.

All items must be unique.

12.1.5 BOM references

Location: /compositions/[]/vulnerabilities
Property: vulnerabilities (Optional)

Type: array (of String)

Description: The bom-ref identifiers of the vulnerabilities being described. Each item of this array must be a string.

All items must be unique.

12.1.6 Signature

Location: /compositions/[]/signature
Property: signature (Optional)

Type: Object

Description: An enveloped digital signature embedded within and specific to this object within the BOM. CycloneDX signatures enable integrity and authenticity verification without separating the signature from the BOM. Enveloped signatures enable each party in the supply chain to take responsibility for and sign their specific data, ensuring its integrity and authenticity. By aggregating all signatures, stakeholders can independently verify discrete pieces of information from each provider, enhancing overall transparency and trust in the supply chain.

Reference: Refer to the JSON Signature Format specification or to the XML Signature specification for implementation details.

- https://cyberphone.github.io/doc/security/jsf.html
- https://www.w3.org/TR/xmldsig-core/

All items must be unique.

13 Vulnerabilities

Location: /vulnerabilities
Property: vulnerabilities (Optional)

Type: Array

Description: Vulnerabilities identified in components or services. Each item of this array must be a Vulnerability object.

13.1 Vulnerability

Location: /vulnerabilities/[]

Type: Object

Description: Defines a weakness in a component or service that could be exploited or triggered by a threat source.

Table 228 – Properties for the vulnerabilities object

Property	Type	Requirement	Description
bom-ref	String	Optional	Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
id	String	Optional	The identifier that uniquely identifies the vulnerability.
source	Object	Optional	The source that published the vulnerability.
references	Array	Optional	Zero or more pointers to vulnerabilities that are the equivalent of the vulnerability specified. Often times, the same vulnerability may exist in multiple sources of vulnerability intelligence, but have different identifiers. References provide a way to correlate vulnerabilities across multiple sources of vulnerability intelligence.
ratings	Array	Optional	List of vulnerability ratings
cwes	Array	Optional	List of Common Weaknesses Enumerations (CWEs) codes that describes this vulnerability.
description	String	Optional	A description of the vulnerability as provided by the source.
detail	String	Optional	If available, an in-depth description of the vulnerability as provided by the source organization. Details often include information useful in understanding root cause.
recommendation	String	Optional	Recommendations of how the vulnerability can be remediated or mitigated.
workaround	String	Optional	A bypass, usually temporary, of the vulnerability that reduces its likelihood and/or impact. Workarounds often involve changes to configuration or deployments.
proofOfConcept	Object	Optional	Evidence used to reproduce the vulnerability.
advisories	Array	Optional	Published advisories of the vulnerability if provided.
created	String	Optional	The date and time (timestamp) when the vulnerability record was created in the vulnerability database.
published	String	Optional	The date and time (timestamp) when the vulnerability record was first published.
updated	String	Optional	The date and time (timestamp) when the vulnerability record was last updated.
rejected	String	Optional	The date and time (timestamp) when the vulnerability record was rejected (if applicable).
credits	Object	Optional	Individuals or organizations credited with the discovery of the vulnerability.
tools	Array	Optional	The tool(s) used to identify, confirm, or score the vulnerability.

Property	Type	Requirement	Description
analysis	Object	Optional	An assessment of the impact and exploitability of the vulnerability.
affects	Array	Optional	The components or services that are affected by the vulnerability.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

13.1.1 BOM Reference

Location: /vulnerabilities/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

13.1.2 ID

Location: /vulnerabilities/[]/id
Property: id (Optional)

Type: String

Description: The identifier that uniquely identifies the vulnerability.

Examples:

- CVE-2021-39182
- GHSA-35m5-8cvj-8783
- SNYK-PYTHON-ENROCRYPT-1912876

13.1.3 Source

Location: /vulnerabilities/[]/source
Property: source (Optional)

Type: Object

Description: The source that published the vulnerability.

Table 229 – Properties for the source object

Property	Type	Requirement	Description
url	String	Optional	The url of the vulnerability documentation as provided by the source.
name	String	Optional	The name of the source.

13.1.3.1 URL

Location: /vulnerabilities/[]/source/url
Property: url (Optional)

Type: String

Description: The url of the vulnerability documentation as provided by the source.

Examples:

- https://nvd.nist.gov/vuln/detail/CVE-2021-39182

13.1.3.2 Name

Location: /vulnerabilities/[]/source/name
Property: name (Optional)

Type: String

Description: The name of the source.

Examples:

- NVD
- National Vulnerability Database
- OSS Index
- VulnDB
- GitHub Advisories

13.1.4 References

Location: /vulnerabilities/[]/references
Property: references (Optional)

Type: Array

Description: Zero or more pointers to vulnerabilities that are the equivalent of the vulnerability specified. Often times, the same vulnerability may exist in multiple sources of vulnerability intelligence, but have different identifiers. References provide a way to correlate vulnerabilities across multiple sources of vulnerability intelligence.

Location: /vulnerabilities/[]/references/[]

Type: Object

Table 230 – Properties for the references object

Property	Type	Requirement	Description
id	String	Required	An identifier that uniquely identifies the vulnerability.
source	Object	Required	The source that published the vulnerability.

13.1.4.1.1 ID

Location: /vulnerabilities/[]/references/[]/id
Property: id (Required)

Type: String

Description: An identifier that uniquely identifies the vulnerability.

Examples:

- CVE-2021-39182
- GHSA-35m5-8cvj-8783
- SNYK-PYTHON-ENROCRYPT-1912876

13.1.4.1.2 Source

Location: /vulnerabilities/[]/references/[]/source
Property: source (Required)

Type: Object

Description: The source that published the vulnerability.

Table 231 – Properties for the source object

Property	Type	Requirement	Description
url	String	Optional	The url of the vulnerability documentation as provided by the source.
name	String	Optional	The name of the source.

13.1.4.1.3 URL

Location: /vulnerabilities/[]/references/[]/source/url
Property: url (Optional)

Type: String

Description: The url of the vulnerability documentation as provided by the source.

Examples:

- https://nvd.nist.gov/vuln/detail/CVE-2021-39182

13.1.4.1.4 Name

Location: /vulnerabilities/[]/references/[]/source/name
Property: name (Optional)

Type: String

Description: The name of the source.

Examples:

- NVD
- National Vulnerability Database
- OSS Index
- VulnDB
- GitHub Advisories

13.1.5 Ratings

Location: /vulnerabilities/[]/ratings
Property: ratings (Optional)

Type: Array

Description: List of vulnerability ratings Each item of this array must be a Rating object.

13.1.5.1 Rating

Location: /vulnerabilities/[]/ratings/[]

Type: Object

Description: Defines the severity or risk ratings of a vulnerability.

Table 232 – Properties for the ratings object

Property	Type	Requirement	Description
source	Object	Optional	The source that calculated the severity or risk rating of the vulnerability.
score	Number	Optional	The numerical score of the rating.
severity	String	Optional	Textual representation of the severity that corresponds to the numerical score of the rating.
method	String	Optional	Specifies the severity or risk scoring methodology or standard used.
vector	String	Optional	Textual representation of the metric values used to score the vulnerability
justification	String	Optional	An optional reason for rating the vulnerability as it was

13.1.5.1.1 Source

Location: /vulnerabilities/[]/ratings/[]/source

Property: source (Optional)

Type: Object

Description: The source that calculated the severity or risk rating of the vulnerability.

Table 233 – Properties for the source object

Property	Type	Requirement	Description
url	String	Optional	The url of the vulnerability documentation as provided by the source.
name	String	Optional	The name of the source.

13.1.5.1.2 URL

Location: /vulnerabilities/[]/ratings/[]/source/url

Property: url (Optional)

Type: String

Description: The url of the vulnerability documentation as provided by the source.

Examples:

- https://nvd.nist.gov/vuln/detail/CVE-2021-39182

13.1.5.1.3 Name

Location: /vulnerabilities/[]/ratings/[]/source/name

Property: name (Optional)

Type: String

Description: The name of the source.

Examples:

- NVD
- National Vulnerability Database
- OSS Index
- VulnDB
- GitHub Advisories

13.1.5.1.4 Score

Location: /vulnerabilities/[]/ratings/[]/score
Property: score (Optional)

Type: Number

Description: The numerical score of the rating.

13.1.5.1.5 Severity

Location: /vulnerabilities/[]/ratings/[]/severity
Property: severity (Optional)

Type: String

Description: Textual representation of the severity that corresponds to the numerical score of the rating.

Table 234 – Enumeration of possible values

Value	Description
critical	Critical severity
high	High severity
medium	Medium severity
low	Low severity
info	Informational warning.
none	None
unknown	The severity is not known

13.1.5.1.6 Method

Location: /vulnerabilities/[]/ratings/[]/method
Property: method (Optional)

Type: String

Description: Specifies the severity or risk scoring methodology or standard used.

Table 235 – Enumeration of possible values

Value	Description
CVSSv2	Common Vulnerability Scoring System v2.0
CVSSv3	Common Vulnerability Scoring System v3.0
CVSSv31	Common Vulnerability Scoring System v3.1
CVSSv4	Common Vulnerability Scoring System v4.0
OWASP	OWASP Risk Rating Methodology
SSVC	Stakeholder Specific Vulnerability Categorization
other	Another severity or risk scoring methodology

13.1.5.1.7 Vector

Location: /vulnerabilities/[]/ratings/[]/vector
Property: vector (Optional)

Type: String

Description: Textual representation of the metric values used to score the vulnerability

13.1.5.1.8 Justification

Location: /vulnerabilities/[]/ratings/[]/justification
Property: justification (Optional)

Type: String

Description: An optional reason for rating the vulnerability as it was

13.1.6 CWEs

Location: /vulnerabilities/[]/cwes
Property: cwes (Optional)

Type: array (of Integer)

Minimum Value: 1

Description: List of Common Weaknesses Enumerations (CWEs) codes that describes this vulnerability. Integer representation of a Common Weaknesses Enumerations (CWE). For example 399 (of https://cwe.mitre.org/data/definitions/399.html) Each item of this array must be an integer.

Examples:

- 399

13.1.7 Description

Location: /vulnerabilities/[]/description
Property: description (Optional)

Type: String

Description: A description of the vulnerability as provided by the source.

13.1.8 Details

Location: /vulnerabilities/[]/detail
Property: detail (Optional)

Type: String

Description: If available, an in-depth description of the vulnerability as provided by the source organization. Details often include information useful in understanding root cause.

13.1.9 Recommendation

Location: /vulnerabilities/[]/recommendation
Property: recommendation (Optional)

Type: String

Description: Recommendations of how the vulnerability can be remediated or mitigated.

13.1.10 Workarounds

Location: /vulnerabilities/[]/workaround
Property: workaround (Optional)

Type: String

Description: A bypass, usually temporary, of the vulnerability that reduces its likelihood and/or impact. Workarounds often involve changes to configuration or deployments.

13.1.11 Proof of Concept

Location: /vulnerabilities/[]/proofOfConcept
Property: proofOfConcept (Optional)

Type: Object

Description: Evidence used to reproduce the vulnerability.

Table 236 – Properties for the proofOfConcept object

Property	Type	Requirement	Description
reproductionSteps	String	Optional	Precise steps to reproduce the vulnerability.
environment	String	Optional	A description of the environment in which reproduction was possible.
supportingMaterial	Array	Optional	Supporting material that helps in reproducing or understanding how reproduction is possible. This may include screenshots, payloads, and PoC exploit code.

13.1.11.1 Steps to Reproduce

Location: /vulnerabilities/[]/proofOfConcept/reproductionSteps
Property: reproductionSteps (Optional)

Type: String

Description: Precise steps to reproduce the vulnerability.

13.1.11.2 Environment

Location: /vulnerabilities/[]/proofOfConcept/environment
Property: environment (Optional)

Type: String

Description: A description of the environment in which reproduction was possible.

13.1.11.3 Supporting Material

Location: /vulnerabilities/[]/proofOfConcept/supportingMaterial
Property: supportingMaterial (Optional)

Type: Array

Description: Supporting material that helps in reproducing or understanding how reproduction is possible. This may include screenshots, payloads, and PoC exploit code. Each item of this array must be an Attachment object.

13.1.11.3.1 Attachment

Location: /vulnerabilities/[]/proofOfConcept/supportingMaterial/[]

Type: Object

Description: Specifies the metadata and content for an attachment.

Table 237 – Properties for the supportingMaterial object

Property	**Type**	**Requirement**	**Description**
contentType	String	Optional	Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.
encoding	String	Optional	Specifies the optional encoding the text is represented in.
content	String	Required	The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

13.1.11.3.2 Content-Type

Location: /vulnerabilities/[]/proofOfConcept/supportingMaterial/[]/contentType
Property: contentType (Optional)

Type: String

Default Value: text/plain

Description: Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.

Examples:

- text/plain
- application/json
- image/png

13.1.11.3.3 Encoding

Location: /vulnerabilities/[]/proofOfConcept/supportingMaterial/[]/encoding
Property: encoding (Optional)

Type: String

Description: Specifies the optional encoding the text is represented in.

Table 238 – Enumeration of possible values

Value	Description
base64	Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string.

13.1.11.3.4 Attachment Text

Location: /vulnerabilities/[]/proofOfConcept/supportingMaterial/[]/content
Property: content (Required)

Type: String

Description: The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

13.1.12 Advisories

Location: /vulnerabilities/[]/advisories
Property: advisories (Optional)

Type: Array

Description: Published advisories of the vulnerability if provided. Each item of this array must be an Advisory object.

13.1.12.1 Advisory

Location: /vulnerabilities/[]/advisories/[]

Type: Object

Description: Title and location where advisory information can be obtained. An advisory is a notification of a threat to a component, service, or system.

Table 239 – Properties for the advisories object

Property	Type	Requirement	Description
title	String	Optional	An optional name of the advisory.
url	String	Required	Location where the advisory can be obtained.

13.1.12.1.1 Title

Location: /vulnerabilities/[]/advisories/[]/title
Property: title (Optional)

Type: String

Description: An optional name of the advisory.

13.1.12.1.2 URL

Location: /vulnerabilities/[]/advisories/[]/url
Property: url (Required)

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: Location where the advisory can be obtained.

13.1.13 Created

Location: /vulnerabilities/[]/created
Property: created (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time (timestamp) when the vulnerability record was created in the vulnerability database.

13.1.14 Published

Location: /vulnerabilities/[]/published
Property: published (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time (timestamp) when the vulnerability record was first published.

13.1.15 Updated

Location: /vulnerabilities/[]/updated
Property: updated (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time (timestamp) when the vulnerability record was last updated.

13.1.16 Rejected

Location: /vulnerabilities/[]/rejected
Property: rejected (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time (timestamp) when the vulnerability record was rejected (if applicable).

13.1.17 Credits

Location: /vulnerabilities/[]/credits
Property: credits (Optional)

Type: Object

Description: Individuals or organizations credited with the discovery of the vulnerability.

Table 240 – Properties for the credits object

Property	Type	Requirement	Description
organizations	Array	Optional	The organizations credited with vulnerability discovery.
individuals	Array	Optional	The individuals, not associated with organizations, that are credited with vulnerability discovery.

13.1.17.1 Organizations

Location: /vulnerabilities/[]/credits/organizations
Property: organizations (Optional)

Type: Array

Description: The organizations credited with vulnerability discovery. Each item of this array must be an Organizational Entity object.

13.1.17.1.1 Organizational Entity

Location: /vulnerabilities/[]/credits/organizations/[]

Type: Object

Table 241 – Properties for the organizations object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

13.1.17.1.2 BOM Reference

Location: /vulnerabilities/[]/credits/organizations/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

13.1.17.1.3 Organization Name

Location: /vulnerabilities/[]/credits/organizations/[]/name
Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

13.1.17.1.4 Organization Address

Location: /vulnerabilities/[]/credits/organizations/[]/address
Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 242 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

13.1.17.1.5 BOM Reference

Location: /vulnerabilities/[]/credits/organizations/[]/address/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

13.1.17.1.6 Country

Location: /vulnerabilities/[]/credits/organizations/[]/address/country
Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

13.1.17.1.7 Region

Location: /vulnerabilities/[]/credits/organizations/[]/address/region
Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

13.1.17.1.8 Locality

Location: /vulnerabilities/[]/credits/organizations/[]/address/locality
Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

13.1.17.1.9 Post Office Box Number

Location: /vulnerabilities/[]/credits/organizations/[]/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

13.1.17.1.10 Postal Code

Location: /vulnerabilities/[]/credits/organizations/[]/address/postalCode
Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

13.1.17.1.11 Street Address

Location: /vulnerabilities/[]/credits/organizations/[]/address/streetAddress
Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

13.1.17.1.12 Organization URL(s)

Location: /vulnerabilities/[]/credits/organizations/[]/url
Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

13.1.17.1.13 Organizational Contact

Location: /vulnerabilities/[]/credits/organizations/[]/contact
Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

13.1.17.1.14 Organizational Contact

Location: /vulnerabilities/[]/credits/organizations/[]/contact/[]

Type: Object

Table 243 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

13.1.17.1.15 BOM Reference

Location: /vulnerabilities/[]/credits/organizations/[]/contact/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

13.1.17.1.16 Name

Location: /vulnerabilities/[]/credits/organizations/[]/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

13.1.17.1.17 Email Address

Location: /vulnerabilities/[]/credits/organizations/[]/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

13.1.17.1.18 Phone

Location: /vulnerabilities/[]/credits/organizations/[]/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

13.1.17.2 Individuals

Location: /vulnerabilities/[]/credits/individuals
Property: individuals (Optional)

Type: Array

Description: The individuals, not associated with organizations, that are credited with vulnerability discovery. Each item of this array must be an Organizational Contact object.

13.1.17.2.1 Organizational Contact

Location: /vulnerabilities/[]/credits/individuals/[]

Type: Object

Table 244 – Properties for the individuals object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

13.1.17.2.2 BOM Reference

Location: /vulnerabilities/[]/credits/individuals/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

13.1.17.2.3 Name

Location: /vulnerabilities/[]/credits/individuals/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

13.1.17.2.4 Email Address

Location: /vulnerabilities/[]/credits/individuals/[]/email

Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

13.1.17.2.5 Phone

Location: /vulnerabilities/[]/credits/individuals/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

13.1.18 Tools

Location: /vulnerabilities/[]/tools
Property: tools (Optional)

Description: The tool(s) used to identify, confirm, or score the vulnerability.

*Must be **one of**:*

1. Tools
2. Tools (legacy)

13.1.19 Tools

Type: Object
Description: The tool(s) used to identify, confirm, or score the vulnerability.

Table 245 – Properties for tools

Property	Type	Requirement	Description
components	Array	Optional	A list of software and hardware components used as tools. Refer to the component definition at /components/[]
services	Array	Optional	A list of services used as tools. This may include microservices, function-as-a-service, and other types of network or intra-process services. Refer to the service definition at /services/[]

13.1.20 Tools (legacy)

Type: Array
Description: [Deprecated] The tool(s) used to identify, confirm, or score the vulnerability.

13.1.20.1 Components

Location: /vulnerabilities/[]/tools/components
Property: tools (Optional)

Type: Array

Description: A list of software and hardware components used as tools. Each item of this array must be a Component object.

13.1.20.1.1 Component

Location: /vulnerabilities/[]/tools/components/[]

Type: Object

Reference: Refer to the component definition at /components/[]

All items must be unique.

13.1.20.2 Services

Location: /vulnerabilities/[]/tools/services

Property: tools (Optional)

Type: Array

Description: A list of services used as tools. This may include microservices, function-as-a-service, and other types of network or intra-process services. Each item of this array must be a Service object.

13.1.20.2.1 Service

Location: /vulnerabilities/[]/tools/services/[]

Type: Object

Reference: Refer to the service definition at /services/[]

All items must be unique.

13.1.21 Tools (legacy)

Location: /vulnerabilities/[]/tools

Property: tools

Type: Array

Description: [Deprecated] The tool(s) used to identify, confirm, or score the vulnerability. Each item of this array must be a Tool object.

13.1.21.1 Tool

Location: /vulnerabilities/[]/tools/[]

Type: Object

Description: [Deprecated] This will be removed in a future version. Use component or service instead. Information about the automated or manual tool used

Table 246 – Properties for the tools object

Property	Type	Requirement	Description
vendor	String	Optional	The name of the vendor who created the tool
name	String	Optional	The name of the tool
version	String	Optional	The version of the tool
hashes	Array	Optional	The hashes of the tool (if applicable).
externalReferences	Array	Optional	External references provide a way to document systems, sites, and information that may be relevant, but are not included with the BOM.

Property	Type	Requirement	Description
			They may also establish specific relationships within or external to the BOM.

13.1.21.1.1 Tool Vendor

Location: /vulnerabilities/[]/tools/[]/vendor
Property: vendor (Optional)

Type: String

Description: The name of the vendor who created the tool

13.1.21.1.2 Tool Name

Location: /vulnerabilities/[]/tools/[]/name
Property: name (Optional)

Type: String

Description: The name of the tool

13.1.21.1.3 Tool Version

Location: /vulnerabilities/[]/tools/[]/version
Property: version (Optional)

Type: String

Description: The version of the tool

Examples:

- 9.0.14
- v1.33.7
- 7.0.0-M1
- 2.0pre1
- 1.0.0-beta1
- 0.8.15

13.1.21.1.4 Hashes

Location: /vulnerabilities/[]/tools/[]/hashes
Property: hashes (Optional)

Type: Array

Description: The hashes of the tool (if applicable). Each item of this array must be a Hash object.

13.1.21.1.5 Hash

Location: /vulnerabilities/[]/tools/[]/hashes/[]

Type: Object

Table 247 – Properties for the hashes object

Property	Type	Requirement	Description
alg	String	Required	The algorithm that generated the hash value.
content	String	Required	The value of the hash.

13.1.21.1.6 Hash Algorithm

Location: /vulnerabilities/[]/tools/[]/hashes/[]/alg
Property: alg (Required)

Type: String

Description: The algorithm that generated the hash value.

Enumeration: Must be one of:

- MD5
- SHA-1
- SHA-256
- SHA-384
- SHA-512
- SHA3-256
- SHA3-384
- SHA3-512
- BLAKE2b-256
- BLAKE2b-384
- BLAKE2b-512
- BLAKE3

13.1.21.1.7 Hash Value

Location: /vulnerabilities/[]/tools/[]/hashes/[]/content
Property: content (Required)

Type: String

Pattern Constraint: ^([a-fA-F0-9]{32}|[a-fA-F0-9]{40}|[a-fA-F0-9]{64}|[a-fA-F0-9]{96}|[a-fA-F0-9]{128})$

Description: The value of the hash.

Examples:

- 3942447fac867ae5cdb3229b658f4d48

13.1.21.1.8 External References

Location: /vulnerabilities/[]/tools/[]/externalReferences
Property: externalReferences (Optional)

Type: Array

Description: External references provide a way to document systems, sites, and information that may be relevant, but are not included with the BOM. They may also establish specific relationships within or external to the BOM. Each item of this array must be an External Reference object.

Reference: Refer to the external reference definition at /externalReferences/[]

13.1.21.1.9 External Reference

Location: /vulnerabilities/[]/tools/[]/externalReferences/[]

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

13.1.22 Impact Analysis

Location: /vulnerabilities/[]/analysis
Property: analysis (Optional)

Type: Object

Description: An assessment of the impact and exploitability of the vulnerability.

Table 248 – Properties for the analysis object

Property	Type	Requirement	Description
state	String	Optional	Declares the current state of an occurrence of a vulnerability, after automated or manual analysis.
justification	String	Optional	The rationale of why the impact analysis state was asserted.
response	Array	Optional	A response to the vulnerability by the manufacturer, supplier, or project responsible for the affected component or service. More than one response is allowed. Responses are strongly encouraged for vulnerabilities where the analysis state is exploitable.
detail	String	Optional	Detailed description of the impact including methods used during assessment. If a vulnerability is not exploitable, this field should include specific details on why the component or service is not impacted by this vulnerability.
firstIssued	String	Optional	The date and time (timestamp) when the analysis was first issued.
lastUpdated	String	Optional	The date and time (timestamp) when the analysis was last updated.

13.1.22.1 Impact Analysis State

Location: /vulnerabilities/[]/analysis/state
Property: state (Optional)

Type: String

Description: Declares the current state of an occurrence of a vulnerability, after automated or manual analysis.

Table 249 – Enumeration of possible values

Value	Description
resolved	The vulnerability has been remediated.
resolved_with_pedigree	The vulnerability has been remediated and evidence of the changes are provided in the affected components pedigree containing verifiable commit history and/or diff(s).

Value	Description
exploitable	The vulnerability may be directly or indirectly exploitable.
in_triage	The vulnerability is being investigated.
false_positive	The vulnerability is not specific to the component or service and was falsely identified or associated.
not_affected	The component or service is not affected by the vulnerability. Justification should be specified for all not_affected cases.

13.1.22.2 Impact Analysis Justification

Location: /vulnerabilities/[]/analysis/justification
Property: justification (Optional)

Type: String

Description: The rationale of why the impact analysis state was asserted.

Table 250 – Enumeration of possible values

Value	Description
code_not_present	The code has been removed or tree-shaked.
code_not_reachable	The vulnerable code is not invoked at runtime.
requires_configuration	Exploitability requires a configurable option to be set/unset.
requires_dependency	Exploitability requires a dependency that is not present.
requires_environment	Exploitability requires a certain environment which is not present.
protected_by_compiler	Exploitability requires a compiler flag to be set/unset.
protected_at_runtime	Exploits are prevented at runtime.
protected_at_perimeter	Attacks are blocked at physical, logical, or network perimeter.
protected_by_mitigating_control	Preventative measures have been implemented that reduce the likelihood and/or impact of the vulnerability.

13.1.22.3 Response

Location: /vulnerabilities/[]/analysis/response
Property: response (Optional)

Type: array (of String)

Description: A response to the vulnerability by the manufacturer, supplier, or project responsible for the affected component or service. More than one response is allowed. Responses are strongly encouraged for vulnerabilities where the analysis state is exploitable. Each item of this array must be a string.

Table 251 – Enumeration of possible values

Value	Description
can_not_fix	Can not fix
will_not_fix	Will not fix
update	Update to a different revision or release
rollback	Revert to a previous revision or release
workaround_available	There is a workaround available

13.1.22.4 Detail

Location: /vulnerabilities/[]/analysis/detail
Property: detail (Optional)

Type: String

Description: Detailed description of the impact including methods used during assessment. If a vulnerability is not exploitable, this field should include specific details on why the component or service is not impacted by this vulnerability.

13.1.22.5 First Issued

Location: /vulnerabilities/[]/analysis/firstIssued
Property: firstIssued (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time (timestamp) when the analysis was first issued.

13.1.22.6 Last Updated

Location: /vulnerabilities/[]/analysis/lastUpdated
Property: lastUpdated (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time (timestamp) when the analysis was last updated.

13.1.23 Affects

Location: /vulnerabilities/[]/affects
Property: affects (Optional)

Type: Array

Description: The components or services that are affected by the vulnerability.

Location: /vulnerabilities/[]/affects/[]

Type: Object

Table 252 – Properties for the affects object

Property	Type	Requirement	Description
ref	Array	Required	References a component or service by the objects bom-ref
versions	Array	Optional	Zero or more individual versions or range of versions.

13.1.23.1.1 Reference

Location: /vulnerabilities/[]/affects/[]/ref
Property: ref (Required)

Description: References a component or service by the objects bom-ref

*Must be **any of**:*

1. Ref
2. BOM-Link Element

13.1.23.1.2 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

13.1.23.1.3 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

13.1.23.1.4 Versions

Location: /vulnerabilities/[]/affects/[]/versions
Property: versions (Optional)

Type: Array

Description: Zero or more individual versions or range of versions.

Location: /vulnerabilities/[]/affects/[]/versions/[]

Type: Object

Table 253 – Properties for the versions object

Property	Type	Requirement	Description
version	String	Optional	A single version of a component or service.
range	String	Optional	A version range specified in Package URL Version Range syntax (vers) which is defined at https://github.com/package-url/purl-spec/VERSION-RANGE-SPEC.rst
status	String	Optional	The vulnerability status for the version or range of versions.

13.1.23.1.5 Version

Location: /vulnerabilities/[]/affects/[]/versions/[]/version
Property: version (Optional)

Type: String

Description: A single version of a component or service.

Examples:

- 9.0.14
- v1.33.7
- 7.0.0-M1
- 2.0pre1
- 1.0.0-beta1
- 0.8.15

13.1.23.1.6 Version Range

Location: /vulnerabilities/[]/affects/[]/versions/[]/range
Property: range (Optional)

Type: String

Description: A version range specified in Package URL Version Range syntax (vers) which is defined at https://github.com/package-url/purl-spec/VERSION-RANGE-SPEC.rst

Examples:

- vers:cargo/9.0.14
- vers:npm/1.2.3|>=2.0.0|<5.0.0
- vers:pypi/0.0.0|0.0.1|0.0.2|0.0.3|1.0|2.0pre1
- vers:tomee/>=1.0.0-beta1|<=1.7.5|>=7.0.0-M1|<=7.0.7|>=7.1.0|<=7.1.2|>=8.0.0-M1|<=8.0.1
- vers:gem/>=2.2.0|!= 2.2.1|<2.3.0

13.1.23.1.7 Status

Location: /vulnerabilities/[]/affects/[]/versions/[]/status
Property: status (Optional)

Type: String

Default Value: affected

Description: The vulnerability status for the version or range of versions.

Table 254 – Enumeration of possible values

Value	Description
affected	The version is affected by the vulnerability.
unaffected	The version is not affected by the vulnerability.
unknown	It is unknown (or unspecified) whether the given version is affected.

All items must be unique.

13.1.24 Properties

Location: /vulnerabilities/[]/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

13.1.24.1 Lightweight name-value pair

Location: /vulnerabilities/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 255 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

13.1.24.1.1 Name

Location: /vulnerabilities/[]/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

13.1.24.1.2 Value

Location: /vulnerabilities/[]/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

All items must be unique.

14 Annotations

Location: /annotations
Property: annotations (Optional)

Type: Array

Description: Comments made by people, organizations, or tools about any object with a bom-ref, such as components, services, vulnerabilities, or the BOM itself. Unlike inventory information, annotations may contain opinions or commentary from various stakeholders. Annotations may be inline (with inventory) or externalized via BOM-Link and may optionally be signed. Each item of this array must be an Annotations object.

14.1 Annotations

Location: /annotations/[]

Type: Object

Description: A comment, note, explanation, or similar textual content which provides additional context to the object(s) being annotated.

Table 256 – Properties for the annotations object

Property	Type	Requirement	Description
bom-ref	String	Optional	Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
subjects	Array	Required	The object in the BOM identified by its bom-ref. This is often a component or service, but may be any object type supporting bom-refs.
annotator	Array	Required	The organization, person, component, or service which created the textual content of the annotation.
timestamp	String	Required	The date and time (timestamp) when the annotation was created.
text	String	Required	The textual content of the annotation.
signature	Array	Optional	Enveloped signature in JSON Signature Format (JSF).

14.1.1 BOM Reference

Location: /annotations/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

14.1.2 Subjects

Location: /annotations/[]/subjects
Property: subjects (Required)

Type: Array

Description: The object in the BOM identified by its bom-ref. This is often a component or service, but may be any object type supporting bom-refs.

Location: /annotations/[]/subjects/[]

*Must be **any of**:*

1. Ref
2. BOM-Link Element

14.1.2.1 Ref

Type: String

Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

14.1.2.2 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

All items must be unique.

14.1.3 Annotator

Location: /annotations/[]/annotator
Property: annotator (Required)

Type: Object

Description: The organization, person, component, or service which created the textual content of the annotation.

Table 257 – Properties for the annotator object

Property	Type	Requirement	Description
organization	Object	Optional	The organization that created the annotation
individual	Object	Optional	The person that created the annotation
component	Object	Optional	The tool or component that created the annotation
service	Object	Optional	The service that created the annotation

14.1.3.1 Organizational Entity

Location: /annotations/[]/annotator/organization
Property: organization (Optional)

Type: Object

Description: The organization that created the annotation

Table 258 – Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization

Property	Type	Requirement	Description
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

14.1.3.1.1 BOM Reference

Location: /annotations/[]/annotator/organization/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

14.1.3.1.2 Organization Name

Location: /annotations/[]/annotator/organization/name
Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

14.1.3.1.3 Organization Address

Location: /annotations/[]/annotator/organization/address
Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 259 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

14.1.3.1.4 BOM Reference

Location: /annotations/[]/annotator/organization/address/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

14.1.3.1.5 Country

Location: /annotations/[]/annotator/organization/address/country
Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

14.1.3.1.6 Region

Location: /annotations/[]/annotator/organization/address/region
Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

14.1.3.1.7 Locality

Location: /annotations/[]/annotator/organization/address/locality
Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

14.1.3.1.8 Post Office Box Number

Location: /annotations/[]/annotator/organization/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

14.1.3.1.9 Postal Code

Location: /annotations/[]/annotator/organization/address/postalCode
Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

14.1.3.1.10 Street Address

Location: /annotations/[]/annotator/organization/address/streetAddress
Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

14.1.3.1.11 Organization URL(s)

Location: /annotations/[]/annotator/organization/url
Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

14.1.3.1.12 Organizational Contact

Location: /annotations/[]/annotator/organization/contact
Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

14.1.3.1.13 Organizational Contact

Location: /annotations/[]/annotator/organization/contact/[]

Type: Object

Table 260 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

14.1.3.1.14 BOM Reference

Location: /annotations/[]/annotator/organization/contact/[]/bom-ref

Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

14.1.3.1.15 Name

Location: /annotations/[]/annotator/organization/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

14.1.3.1.16 Email Address

Location: /annotations/[]/annotator/organization/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

14.1.3.1.17 Phone

Location: /annotations/[]/annotator/organization/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

14.1.3.2 Organizational Contact

Location: /annotations/[]/annotator/individual
Property: individual (Optional)

Type: Object

Description: The person that created the annotation

Table 261 – Properties for the individual object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

14.1.3.2.1 BOM Reference

Location: /annotations/[]/annotator/individual/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

14.1.3.2.2 Name

Location: /annotations/[]/annotator/individual/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

14.1.3.2.3 Email Address

Location: /annotations/[]/annotator/individual/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

14.1.3.2.4 Phone

Location: /annotations/[]/annotator/individual/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

14.1.3.3 Component

Location: /annotations/[]/annotator/component
Property: component (Optional)

Type: Object

Description: The tool or component that created the annotation

Reference: Refer to the component definition at /components/[]

14.1.3.4 Service

Location: /annotations/[]/annotator/service
Property: service (Optional)

Type: Object

Description: The service that created the annotation

Reference: Refer to the service definition at /services/[]

14.1.4 Timestamp

Location: /annotations/[]/timestamp
Property: timestamp (Required)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time (timestamp) when the annotation was created.

14.1.5 Text

Location: /annotations/[]/text
Property: text (Required)

Type: String

Description: The textual content of the annotation.

14.1.6 Signature

Location: /annotations/[]/signature
Property: signature (Optional)

Type: Object

Description: An enveloped digital signature embedded within and specific to this object within the BOM. CycloneDX signatures enable integrity and authenticity verification without separating the signature from the BOM. Enveloped signatures enable each party in the supply chain to take responsibility for and sign their specific data, ensuring its integrity and authenticity. By aggregating all signatures, stakeholders can independently verify discrete pieces of information from each provider, enhancing overall transparency and trust in the supply chain.

Reference: Refer to the JSON Signature Format specification or to the XML Signature specification for implementation details.

- https://cyberphone.github.io/doc/security/jsf.html
- https://www.w3.org/TR/xmldsig-core/

All items must be unique.

15 Formulation

Location: /formulation
Property: formulation (Optional)

Type: Array

Description: Describes how a component or service was manufactured or deployed. This is achieved through the use of formulas, workflows, tasks, and steps, which declare the precise steps to reproduce along with the observed formulas describing the steps which transpired in the manufacturing process. Each item of this array must be a Formula object.

15.1 Formula

Location: /formulation/[]

Type: Object

Description: Describes workflows and resources that captures rules and other aspects of how the associated BOM component or service was formed.

Table 262 – Properties for the formulation object

Property	Type	Requirement	Description
bom-ref	String	Optional	Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
components	Array	Optional	Transient components that are used in tasks that constitute one or more of this formula's workflows
services	Array	Optional	Transient services that are used in tasks that constitute one or more of this formula's workflows
workflows	Array	Optional	List of workflows that can be declared to accomplish specific orchestrated goals and independently triggered.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.1 BOM Reference

Location: /formulation/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

15.1.2 Components

Location: /formulation/[]/components
Property: components (Optional)

Type: Array

Description: Transient components that are used in tasks that constitute one or more of this formula's workflows Each item of this array must be a Component object.

15.1.2.1 Component

Location: /formulation/[]/components/[]

Type: Object
Reference: Refer to the component definition at /components/[]

All items must be unique.

15.1.3 Services

Location: /formulation/[]/services
Property: services (Optional)

Type: Array

Description: Transient services that are used in tasks that constitute one or more of this formula's workflows Each item of this array must be a Service object.

15.1.3.1 Service

Location: /formulation/[]/services/[]

Type: Object
Reference: Refer to the service definition at /services/[]

All items must be unique.

15.1.4 Workflows

Location: /formulation/[]/workflows
Property: workflows (Optional)

Type: Array

Description: List of workflows that can be declared to accomplish specific orchestrated goals and independently triggered. Each item of this array must be a Workflow object.

15.1.4.1 Workflow

Location: /formulation/[]/workflows/[]

Type: Object
Description: A specialized orchestration task.

Table 263 – Properties for the workflows object

Property	Type	Requirement	Description
bom-ref	String	Required	Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
uid	String	Required	The unique identifier for the resource instance within its deployment context.
name	String	Optional	The name of the resource instance.
description	String	Optional	A description of the resource instance.
resourceReferences	Array	Optional	References to component or service resources that are used to realize the resource instance.

Property	Type	Requirement	Description
tasks	Array	Optional	The tasks that comprise the workflow.
taskDependencies	Array	Optional	The graph of dependencies between tasks within the workflow.
taskTypes	Array	Required	Indicates the types of activities performed by the set of workflow tasks.
trigger	Object	Optional	The trigger that initiated the task.
steps	Array	Optional	The sequence of steps for the task.
inputs	Array	Optional	Represents resources and data brought into a task at runtime by executor or task commands
outputs	Array	Optional	Represents resources and data output from a task at runtime by executor or task commands
timeStart	String	Optional	The date and time (timestamp) when the task started.
timeEnd	String	Optional	The date and time (timestamp) when the task ended.
workspaces	Array	Optional	A set of named filesystem or data resource shareable by workflow tasks.
runtimeTopology	Array	Optional	A graph of the component runtime topology for workflow's instance.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.1 BOM Reference

Location: /formulation/[]/workflows/[]/bom-ref
Property: bom-ref (Required)

Type: String

Description: Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

15.1.4.1.2 Unique Identifier (UID)

Location: /formulation/[]/workflows/[]/uid
Property: uid (Required)

Type: String

Description: The unique identifier for the resource instance within its deployment context.

15.1.4.1.3 Name

Location: /formulation/[]/workflows/[]/name

Property: name (Optional)

Type: String

Description: The name of the resource instance.

15.1.4.1.4 Description

Location: /formulation/[]/workflows/[]/description
Property: description (Optional)

Type: String

Description: A description of the resource instance.

15.1.4.1.5 Resource references

Location: /formulation/[]/workflows/[]/resourceReferences
Property: resourceReferences (Optional)

Type: Array

Description: References to component or service resources that are used to realize the resource instance. Each item of this array must be a Resource reference choice object.

15.1.4.1.6 Resource reference choice

Location: /formulation/[]/workflows/[]/resourceReferences/[]

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Table 264 – Properties for the resourceReferences object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.7 BOM Reference

Location: /formulation/[]/workflows/[]/resourceReferences/[]/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.8 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.9 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$

Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.10 External Reference

Location: /formulation/[]/workflows/[]/resourceReferences/[]/externalReference

Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

All items must be unique.

15.1.4.1.11 Tasks

Location: /formulation/[]/workflows/[]/tasks

Property: tasks (Optional)

Type: Array

Description: The tasks that comprise the workflow. Each item of this array must be a Task object.

15.1.4.1.12 Task

Location: /formulation/[]/workflows/[]/tasks/[]

Type: Object

Description: Describes the inputs, sequence of steps and resources used to accomplish a task and its output.

Table 265 – Properties for the tasks object

Property	Type	Requirement	Description
bom-ref	String	Required	Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
uid	String	Required	The unique identifier for the resource instance within its deployment context.
name	String	Optional	The name of the resource instance.
description	String	Optional	A description of the resource instance.
resourceReferences	Array	Optional	References to component or service resources that are used to realize the resource instance.
taskTypes	Array	Required	Indicates the types of activities performed by the set of workflow tasks.
trigger	Object	Optional	The trigger that initiated the task.
steps	Array	Optional	The sequence of steps for the task.

Property	Type	Requirement	Description
inputs	Array	Optional	Represents resources and data brought into a task at runtime by executor or task commands
outputs	Array	Optional	Represents resources and data output from a task at runtime by executor or task commands
timeStart	String	Optional	The date and time (timestamp) when the task started.
timeEnd	String	Optional	The date and time (timestamp) when the task ended.
workspaces	Array	Optional	A set of named filesystem or data resource shareable by workflow tasks.
runtimeTopology	Array	Optional	A graph of the component runtime topology for task's instance.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.13 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/bom-ref
Property: bom-ref (Required)

Type: String

Description: Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

15.1.4.1.14 Unique Identifier (UID)

Location: /formulation/[]/workflows/[]/tasks/[]/uid
Property: uid (Required)

Type: String

Description: The unique identifier for the resource instance within its deployment context.

15.1.4.1.15 Name

Location: /formulation/[]/workflows/[]/tasks/[]/name
Property: name (Optional)

Type: String

Description: The name of the resource instance.

15.1.4.1.16 Description

Location: /formulation/[]/workflows/[]/tasks/[]/description
Property: description (Optional)

Type: String

Description: A description of the resource instance.

15.1.4.1.17 Resource references

Location: /formulation/[]/workflows/[]/tasks/[]/resourceReferences

Property: resourceReferences (Optional)

Type: Array

Description: References to component or service resources that are used to realize the resource instance. Each item of this array must be a Resource reference choice object.

15.1.4.1.18 Resource reference choice

Location: /formulation/[]/workflows/[]/tasks/[]/resourceReferences/[]

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Table 266 – Properties for the resourceReferences object

Property	**Type**	**Requirement**	**Description**
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.19 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/resourceReferences/[]/ref

Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.20 Ref

Type: String

Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.21 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$

Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.22 External Reference

Location: /formulation/[]/workflows/[]/tasks/[]/resourceReferences/[]/externalReference

Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

All items must be unique.

15.1.4.1.23 Task types

Location: /formulation/[]/workflows/[]/tasks/[]/taskTypes
Property: taskTypes (Required)

Type: array (of String)

Description: Indicates the types of activities performed by the set of workflow tasks. Each item of this array must be a string.

Table 267 – Enumeration of possible values

Value	Description
copy	A task that copies software or data used to accomplish other tasks in the workflow.
clone	A task that clones a software repository into the workflow in order to retrieve its source code or data for use in a build step.
lint	A task that checks source code for programmatic and stylistic errors.
scan	A task that performs a scan against source code, or built or deployed components and services. Scans are typically run to gather or test for security vulnerabilities or policy compliance.
merge	A task that merges changes or fixes into source code prior to a build step in the workflow.
build	A task that builds the source code, dependencies and/or data into an artefact that can be deployed to and executed on target systems.
test	A task that verifies the functionality of a component or service.
deliver	A task that delivers a built artefact to one or more target repositories or storage systems.
deploy	A task that deploys a built artefact for execution on one or more target systems.
release	A task that releases a built, versioned artefact to a target repository or distribution system.
clean	A task that cleans unnecessary tools, build artifacts and/or data from workflow storage.
other	A workflow task that does not match current task type definitions.

15.1.4.1.24 Trigger

Location: /formulation/[]/workflows/[]/tasks/[]/trigger
Property: trigger (Optional)

Type: Object

Description: The trigger that initiated the task.

Table 268 – Properties for the trigger object

Property	Type	Requirement	Description
bom-ref	String	Required	Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
uid	String	Required	The unique identifier for the resource instance within its deployment context.
name	String	Optional	The name of the resource instance.
description	String	Optional	A description of the resource instance.
resourceReferences	Array	Optional	References to component or service resources that are used to realize the resource instance.
type	String	Required	The source type of event which caused the trigger to fire.
event	Object	Optional	The event data that caused the associated trigger to activate.
conditions	Array	Optional	A list of conditions used to determine if a trigger should be activated.
timeActivated	String	Optional	The date and time (timestamp) when the trigger was activated.
inputs	Array	Optional	Represents resources and data brought into a task at runtime by executor or task commands
outputs	Array	Optional	Represents resources and data output from a task at runtime by executor or task commands
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.25 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/bom-ref
Property: bom-ref (Required)

Type: String

Description: Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

15.1.4.1.26 Unique Identifier (UID)

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/uid
Property: uid (Required)

Type: String

Description: The unique identifier for the resource instance within its deployment context.

15.1.4.1.27 Name

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/name
Property: name (Optional)

Type: String

Description: The name of the resource instance.

15.1.4.1.28 Description

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/description
Property: description (Optional)

Type: String

Description: A description of the resource instance.

15.1.4.1.29 Resource references

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/resourceReferences
Property: resourceReferences (Optional)

Type: Array

Description: References to component or service resources that are used to realize the resource instance. Each item of this array must be a Resource reference choice object.

15.1.4.1.30 Resource reference choice

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/resourceReferences/[]

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Table 269 – Properties for the resourceReferences object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.31 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/resourceReferences/[]/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.32 Ref

Type: String

Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.33 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.34 External Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/resourceReferences/[]/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

All items must be unique.

15.1.4.1.35 Type

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/type
Property: type (Required)

Type: String

Description: The source type of event which caused the trigger to fire.

Enumeration: Must be one of:

- manual
- api
- webhook
- scheduled

15.1.4.1.36 Event

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event
Property: event (Optional)

Type: Object

Description: The event data that caused the associated trigger to activate.

Table 270 – Properties for the event object

Property	Type	Requirement	Description
uid	String	Optional	The unique identifier of the event.
description	String	Optional	A description of the event.
timeReceived	String	Optional	The date and time (timestamp) when the event was received.
data	Object	Optional	Specifies the metadata and content for an attachment.

Property	Type	Requirement	Description
source	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
target	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.37 Unique Identifier (UID)

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/uid
Property: uid (Optional)

Type: String

Description: The unique identifier of the event.

15.1.4.1.38 Description

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/description
Property: description (Optional)

Type: String

Description: A description of the event.

15.1.4.1.39 Time Received

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/timeReceived
Property: timeReceived (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time (timestamp) when the event was received.

15.1.4.1.40 Attachment

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/data
Property: data (Optional)

Type: Object

Description: Specifies the metadata and content for an attachment.

Table 271 – Properties for the data object

Property	Type	Requirement	Description
contentType	String	Optional	Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.
encoding	String	Optional	Specifies the optional encoding the text is represented in.
content	String	Required	The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

15.1.4.1.41 Content-Type

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/data/contentType
Property: contentType (Optional)

Type: String

Default Value: text/plain

Description: Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.

Examples:

- text/plain
- application/json
- image/png

15.1.4.1.42 Encoding

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/data/encoding
Property: encoding (Optional)

Type: String

Description: Specifies the optional encoding the text is represented in.

Table 272 – Enumeration of possible values

Value	Description
base64	Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string.

15.1.4.1.43 Attachment Text

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/data/content
Property: content (Required)

Type: String

Description: The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

15.1.4.1.44 Resource reference choice

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/source
Property: source (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Table 273 – Properties for the source object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.45 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/source/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

Must be ***any of****:*

1. Ref
2. BOM-Link Element

15.1.4.1.46 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.47 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.48 External Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/source/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.49 Resource reference choice

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/target
Property: target (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Table 274 – Properties for the target object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.50 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/target/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.51 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.52 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.53 External Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/target/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.54 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/properties

Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.55 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 275 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.56 Name

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.57 Value

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

15.1.4.1.58 Conditions

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/conditions
Property: conditions (Optional)

Type: Array

Description: A list of conditions used to determine if a trigger should be activated. Each item of this array must be a Condition object.

15.1.4.1.59 Condition

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/conditions/[]

Type: Object

Description: A condition that was used to determine a trigger should be activated.

Table 276 – Properties for the conditions object

Property	Type	Requirement	Description
description	String	Optional	Describes the set of conditions which cause the trigger to activate.
expression	String	Optional	The logical expression that was evaluated that determined the trigger should be fired.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.60 Description

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/conditions/[]/description
Property: description (Optional)

Type: String

Description: Describes the set of conditions which cause the trigger to activate.

15.1.4.1.61 Expression

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/conditions/[]/expression
Property: expression (Optional)

Type: String

Description: The logical expression that was evaluated that determined the trigger should be fired.

15.1.4.1.62 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/conditions/[]/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.63 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/conditions/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 277 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.64 Name

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/conditions/[]/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.65 Value

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/conditions/[]/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.4.1.66 Time activated

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/timeActivated
Property: timeActivated (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time (timestamp) when the trigger was activated.

15.1.4.1.67 Inputs

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs
Property: inputs (Optional)

Type: Array

Description: Represents resources and data brought into a task at runtime by executor or task commands Each item of this array must be an Input type object.

Examples:

- a configurationfile which was declared as a localcomponentorexternalReference``

15.1.4.1.68 Input type

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]

Type: Object

Description: Type that represents various input data types and formats.

Table 278 – Properties for the inputs object

Property	Type	Requirement	Description
source	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
target	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
resource	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
parameters	Array	Optional	Inputs that have the form of parameters with names and values.
environmentVars	Array	Optional	Inputs that have the form of parameters with names and values.
data	Object	Optional	Specifies the metadata and content for an attachment.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.69 Resource reference choice

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/source
Property: source (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Examples:

- source code repository
- database

Table 279 – Properties for the source object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.70 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/source/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.71 Ref

Type: String

Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.72 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$

Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.73 External Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/source/externalReference

Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.74 Resource reference choice

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/target

Property: target (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Examples:

- workspace
- directory

Table 280 – Properties for the target object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.75 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/target/ref

Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.76 Ref

Type: String

Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.77 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$

Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.78 External Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/target/externalReference

Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.79 Resource reference choice

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/resource

Property: resource (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Examples:

- a reference to a configuration file in a repository (i.e., a bom-ref)
- a reference to a scanning service used in a task (i.e., a bom-ref)

Table 281 – Properties for the resource object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.80 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/resource/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.81 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.82 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.83 External Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/resource/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.84 Parameters

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/parameters
Property: parameters (Optional)

Type: Array

Description: Inputs that have the form of parameters with names and values. Each item of this array must be a Parameter object.

15.1.4.1.85 Parameter

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/parameters/[]

Type: Object
Description: A representation of a functional parameter.

Table 282 – Properties for the parameters object

Property	Type	Requirement	Description
name	String	Optional	The name of the parameter.
value	String	Optional	The value of the parameter.
dataType	String	Optional	The data type of the parameter.

15.1.4.1.86 Name

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/parameters/[]/name
Property: name (Optional)

Type: String

Description: The name of the parameter.

15.1.4.1.87 Value

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/parameters/[]/value
Property: value (Optional)

Type: String

Description: The value of the parameter.

15.1.4.1.88 Data type

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/parameters/[]/dataType
Property: dataType (Optional)

Type: String

Description: The data type of the parameter.

All items must be unique.

15.1.4.1.89 Environment variables

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/environmentVars
Property: environmentVars (Optional)

Type: Array

Description: Inputs that have the form of parameters with names and values.

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/environmentVars/[]

*Must be **one of**:*

1. Lightweight name-value pair

15.1.4.1.90 Lightweight name-value pair

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 283 – Properties for lightweight name-value pair

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

Type: String

15.1.4.1.91 Name

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/environmentVars/[]/name
Property: environmentVars (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.92 Value

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/environmentVars/[]/value
Property: environmentVars (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.4.1.93 Attachment

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/data
Property: data (Optional)

Type: Object

Description: Specifies the metadata and content for an attachment.

Table 284 – Properties for the data object

Property	Type	Requirement	Description
contentType	String	Optional	Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.
encoding	String	Optional	Specifies the optional encoding the text is represented in.
content	String	Required	The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

15.1.4.1.94 Content-Type

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/data/contentType
Property: contentType (Optional)

Type: String

Default Value: text/plain

Description: Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.

Examples:

- text/plain
- application/json
- image/png

15.1.4.1.95 Encoding

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/data/encoding
Property: encoding (Optional)

Type: String

Description: Specifies the optional encoding the text is represented in.

Table 285 – Enumeration of possible values

Value	Description
base64	Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string.

15.1.4.1.96 Attachment Text

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/data/content
Property: content (Required)

Type: String

Description: The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

15.1.4.1.97 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.98 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different

values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 286 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.99 Name

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.100 Value

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.4.1.101 Outputs

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs
Property: outputs (Optional)

Type: Array

Description: Represents resources and data output from a task at runtime by executor or task commands

Examples:

- a log file or metrics data produced by the task

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]

Type: Object

Table 287 – Properties for the outputs object

Property	Type	Requirement	Description
type	String	Optional	Describes the type of data output.
source	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
target	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Property	Type	Requirement	Description
resource	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
data	Object	Optional	Specifies the metadata and content for an attachment.
environmentVars	Array	Optional	Outputs that have the form of environment variables.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.102 Type

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/type
Property: type (Optional)

Type: String

Description: Describes the type of data output.

Enumeration: Must be one of:

- artifact
- attestation
- log
- evidence
- metrics
- other

15.1.4.1.103 Resource reference choice

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/source
Property: source (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Table 288 – Properties for the source object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.104 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/source/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.105 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.106 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.107 External Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/source/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.108 Resource reference choice

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/target
Property: target (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Examples:

- a log file described as an externalReference within its target domain.

Table 289 – Properties for the target object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.109 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/target/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.110 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.111 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.112 External Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/target/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.113 Resource reference choice

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/resource
Property: resource (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Examples:

- configuration file
- source code
- scanning service

Table 290 – Properties for the resource object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.114 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/resource/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.115 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.116 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.117 External Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/resource/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.118 Attachment

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/data
Property: data (Optional)

Type: Object

Description: Specifies the metadata and content for an attachment.

Table 291 – Properties for the data object

Property	Type	Requirement	Description
contentType	String	Optional	Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.
encoding	String	Optional	Specifies the optional encoding the text is represented in.
content	String	Required	The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

15.1.4.1.119 Content-Type

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/data/contentType
Property: contentType (Optional)

Type: String

Default Value: text/plain

Description: Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.

Examples:

- text/plain
- application/json
- image/png

15.1.4.1.120 Encoding

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/data/encoding
Property: encoding (Optional)

Type: String

Description: Specifies the optional encoding the text is represented in.

Table 292 – Enumeration of possible values

Value	Description
base64	Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string.

15.1.4.1.121 Attachment Text

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/data/content
Property: content (Required)

Type: String

Description: The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

15.1.4.1.122 Environment variables

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/environmentVars
Property: environmentVars (Optional)

Type: Array

Description: Outputs that have the form of environment variables.

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/environmentVars/[]

*Must be **one of**:*

1. Lightweight name-value pair

15.1.4.1.123 Lightweight name-value pair

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 293 – Properties for lightweight name-value pair

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

Type: String

15.1.4.1.124 Name

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/environmentVars/[]/name
Property: environmentVars (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.125 Value

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/environmentVars/[]/value
Property: environmentVars (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.4.1.126 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.127 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 294 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.128 Name

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.129 Value

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.4.1.130 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.131 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 295 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.132 Name

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.133 Value

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

15.1.4.1.134 Steps

Location: /formulation/[]/workflows/[]/tasks/[]/steps
Property: steps (Optional)

Type: Array

Description: The sequence of steps for the task.

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]

Type: Object

Description: Executes specific commands or tools in order to accomplish its owning task as part of a sequence.

Table 296 – Properties for the steps object

Property	Type	Requirement	Description
name	String	Optional	A name for the step.
description	String	Optional	A description of the step.
commands	Array	Optional	Ordered list of commands or directives for the step

Property	Type	Requirement	Description
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.135 Name

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/name
Property: name (Optional)

Type: String

Description: A name for the step.

15.1.4.1.136 Description

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/description
Property: description (Optional)

Type: String

Description: A description of the step.

15.1.4.1.137 Commands

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/commands
Property: commands (Optional)

Type: Array

Description: Ordered list of commands or directives for the step

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/commands/[]

Type: Object

Table 297 – Properties for the commands object

Property	Type	Requirement	Description
executed	String	Optional	A text representation of the executed command.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.138 Executed

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/commands/[]/executed
Property: executed (Optional)

Type: String

Description: A text representation of the executed command.

15.1.4.1.139 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/commands/[]/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.140 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/commands/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 298 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.141 Name

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/commands/[]/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.142 Value

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/commands/[]/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

15.1.4.1.143 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX

Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.144 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 299 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.145 Name

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.146 Value

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.4.1.147 Inputs

Location: /formulation/[]/workflows/[]/tasks/[]/inputs
Property: inputs (Optional)

Type: Array

Description: Represents resources and data brought into a task at runtime by executor or task commands Each item of this array must be an Input type object.

Examples:

- a configurationfile which was declared as a localcomponentorexternalReference``

15.1.4.1.148 Input type

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]

Type: Object

Description: Type that represents various input data types and formats.

Table 300 – Properties for the inputs object

Property	Type	Requirement	Description
source	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
target	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
resource	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
parameters	Array	Optional	Inputs that have the form of parameters with names and values.
environmentVars	Array	Optional	Inputs that have the form of parameters with names and values.
data	Object	Optional	Specifies the metadata and content for an attachment.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.149 Resource reference choice

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/source
Property: source (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Examples:

- source code repository
- database

Table 301 – Properties for the source object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.150 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/source/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.151 Ref

Type: String

Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.152 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$

Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.153 External Reference

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/source/externalReference

Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.154 Resource reference choice

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/target

Property: target (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Examples:

- workspace
- directory

Table 302 – Properties for the target object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.155 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/target/ref

Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.156 Ref

Type: String

Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.157 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$

Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.158 External Reference

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/target/externalReference

Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.159 Resource reference choice

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/resource

Property: resource (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Examples:

- a reference to a configuration file in a repository (i.e., a bom-ref)
- a reference to a scanning service used in a task (i.e., a bom-ref)

Table 303 – Properties for the resource object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.160 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/resource/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.161 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.162 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.163 External Reference

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/resource/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.164 Parameters

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/parameters
Property: parameters (Optional)

Type: Array

Description: Inputs that have the form of parameters with names and values. Each item of this array must be a Parameter object.

15.1.4.1.165 Parameter

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/parameters/[]

Type: Object
Description: A representation of a functional parameter.

Table 304 – Properties for the parameters object

Property	Type	Requirement	Description
name	String	Optional	The name of the parameter.
value	String	Optional	The value of the parameter.
dataType	String	Optional	The data type of the parameter.

15.1.4.1.166 Name

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/parameters/[]/name
Property: name (Optional)

Type: String

Description: The name of the parameter.

15.1.4.1.167 Value

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/parameters/[]/value
Property: value (Optional)

Type: String

Description: The value of the parameter.

15.1.4.1.168 Data type

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/parameters/[]/dataType
Property: dataType (Optional)

Type: String

Description: The data type of the parameter.

All items must be unique.

15.1.4.1.169 Environment variables

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/environmentVars
Property: environmentVars (Optional)

Type: Array

Description: Inputs that have the form of parameters with names and values.

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/environmentVars/[]

*Must be **one of**:*

1. Lightweight name-value pair

15.1.4.1.170 Lightweight name-value pair

Type: Object
Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 305 – Properties for lightweight name-value pair

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

Type: String

15.1.4.1.171 Name

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/environmentVars/[]/name
Property: environmentVars (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.172 Value

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/environmentVars/[]/value
Property: environmentVars (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.4.1.173 Attachment

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/data
Property: data (Optional)

Type: Object

Description: Specifies the metadata and content for an attachment.

Table 306 – Properties for the data object

Property	Type	Requirement	Description
contentType	String	Optional	Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.
encoding	String	Optional	Specifies the optional encoding the text is represented in.
content	String	Required	The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

15.1.4.1.174 Content-Type

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/data/contentType
Property: contentType (Optional)

Type: String

Default Value: text/plain

Description: Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.

Examples:

- text/plain
- application/json
- image/png

15.1.4.1.175 Encoding

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/data/encoding
Property: encoding (Optional)

Type: String

Description: Specifies the optional encoding the text is represented in.

Table 307 – Enumeration of possible values

Value	Description
base64	Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string.

15.1.4.1.176 Attachment Text

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/data/content
Property: content (Required)

Type: String

Description: The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

15.1.4.1.177 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.178 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different

values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 308 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.179 Name

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.180 Value

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.4.1.181 Outputs

Location: /formulation/[]/workflows/[]/tasks/[]/outputs
Property: outputs (Optional)

Type: Array

Description: Represents resources and data output from a task at runtime by executor or task commands

Examples:

- a log file or metrics data produced by the task

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]

Type: Object

Table 309 – Properties for the outputs object

Property	Type	Requirement	Description
type	String	Optional	Describes the type of data output.
source	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
target	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Property	Type	Requirement	Description
resource	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
data	Object	Optional	Specifies the metadata and content for an attachment.
environmentVars	Array	Optional	Outputs that have the form of environment variables.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.182 Type

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/type
Property: type (Optional)

Type: String

Description: Describes the type of data output.

Enumeration: Must be one of:

- artifact
- attestation
- log
- evidence
- metrics
- other

15.1.4.1.183 Resource reference choice

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/source
Property: source (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Table 310 – Properties for the source object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.184 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/source/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.185 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.186 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.187 External Reference

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/source/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.188 Resource reference choice

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/target
Property: target (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Examples:

- a log file described as an externalReference within its target domain.

Table 311 – Properties for the target object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.189 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/target/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.190 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.191 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.192 External Reference

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/target/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.193 Resource reference choice

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/resource
Property: resource (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Examples:

- configuration file
- source code
- scanning service

Table 312 – Properties for the resource object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM.

Property	Type	Requirement	Description
			They may also establish specific relationships within or external to the BOM.

15.1.4.1.194 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/resource/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.195 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.196 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.197 External Reference

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/resource/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.198 Attachment

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/data
Property: data (Optional)

Type: Object

Description: Specifies the metadata and content for an attachment.

Table 313 – Properties for the data object

Property	Type	Requirement	Description
contentType	String	Optional	Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a

Property	Type	Requirement	Description
			comprehensive list of registered content types, refer to the IANA media types registry.
encoding	String	Optional	Specifies the optional encoding the text is represented in.
content	String	Required	The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

15.1.4.1.199 Content-Type

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/data/contentType
Property: contentType (Optional)

Type: String

Default Value: text/plain

Description: Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.

Examples:

- text/plain
- application/json
- image/png

15.1.4.1.200 Encoding

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/data/encoding
Property: encoding (Optional)

Type: String

Description: Specifies the optional encoding the text is represented in.

Table 314 – Enumeration of possible values

Value	Description
base64	Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string.

15.1.4.1.201 Attachment Text

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/data/content
Property: content (Required)

Type: String

Description: The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

15.1.4.1.202 Environment variables

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/environmentVars
Property: environmentVars (Optional)

Type: Array

Description: Outputs that have the form of environment variables.

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/environmentVars/[]

*Must be **one of**:*

1. Lightweight name-value pair

15.1.4.1.203 Lightweight name-value pair

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 315 – Properties for lightweight name-value pair

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

Type: String

15.1.4.1.204 Name

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/environmentVars/[]/name
Property: environmentVars (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.205 Value

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/environmentVars/[]/value
Property: environmentVars (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.4.1.206 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.207 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 316 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.208 Name

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.209 Value

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.4.1.210 Time start

Location: /formulation/[]/workflows/[]/tasks/[]/timeStart
Property: timeStart (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time (timestamp) when the task started.

15.1.4.1.211 Time end

Location: /formulation/[]/workflows/[]/tasks/[]/timeEnd
Property: timeEnd (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time (timestamp) when the task ended.

15.1.4.1.212 Workspaces

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces

Property: workspaces (Optional)

Type: Array

Description: A set of named filesystem or data resource shareable by workflow tasks. Each item of this array must be a Workspace object.

15.1.4.1.213 Workspace

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]

Type: Object

Description: A named filesystem or data resource shareable by workflow tasks.

Table 317 – Properties for the workspaces object

Property	Type	Requirement	Description
bom-ref	String	Required	Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
uid	String	Required	The unique identifier for the resource instance within its deployment context.
name	String	Optional	The name of the resource instance.
aliases	Array	Optional	The names for the workspace as referenced by other workflow tasks. Effectively, a name mapping so other tasks can use their own local name in their steps.
description	String	Optional	A description of the resource instance.
resourceReferences	Array	Optional	References to component or service resources that are used to realize the resource instance.
accessMode	String	Optional	Describes the read-write access control for the workspace relative to the owning resource instance.
mountPath	String	Optional	A path to a location on disk where the workspace will be available to the associated task's steps.
managedDataType	String	Optional	The name of a domain-specific data type the workspace represents.
volumeRequest	String	Optional	Identifies the reference to the request for a specific volume type and parameters.
volume	Object	Optional	Information about the actual volume instance allocated to the workspace.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.214 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/bom-ref
Property: bom-ref (Required)

Type: String

Description: Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

15.1.4.1.215 Unique Identifier (UID)

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/uid
Property: uid (Required)

Type: String

Description: The unique identifier for the resource instance within its deployment context.

15.1.4.1.216 Name

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/name
Property: name (Optional)

Type: String

Description: The name of the resource instance.

15.1.4.1.217 Aliases

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/aliases
Property: aliases (Optional)

Type: array (of String)

Description: The names for the workspace as referenced by other workflow tasks. Effectively, a name mapping so other tasks can use their own local name in their steps. Each item of this array must be a string.

15.1.4.1.218 Description

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/description
Property: description (Optional)

Type: String

Description: A description of the resource instance.

15.1.4.1.219 Resource references

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/resourceReferences
Property: resourceReferences (Optional)

Type: Array

Description: References to component or service resources that are used to realize the resource instance. Each item of this array must be a Resource reference choice object.

15.1.4.1.220 Resource reference choice

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/resourceReferences/[]

Type: Object
Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Table 318 – Properties for the resourceReferences object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.221 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/resourceReferences/[]/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.222 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.223 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.224 External Reference

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/resourceReferences/[]/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

All items must be unique.

15.1.4.1.225 Access mode

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/accessMode
Property: accessMode (Optional)

Type: String

Description: Describes the read-write access control for the workspace relative to the owning resource instance.

Enumeration: Must be one of:

- read-only

- read-write
- read-write-once
- write-once
- write-only

15.1.4.1.226 Mount path

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/mountPath
Property: mountPath (Optional)

Type: String

Description: A path to a location on disc where the workspace will be available to the associated task's steps.

15.1.4.1.227 Managed data type

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/managedDataType
Property: managedDataType (Optional)

Type: String

Description: The name of a domain-specific data type the workspace represents.

Examples:

- ConfigMap
- Secret

15.1.4.1.228 Volume request

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volumeRequest
Property: volumeRequest (Optional)

Type: String

Description: Identifies the reference to the request for a specific volume type and parameters.

Examples:

- a kubernetes Persistent Volume Claim (PVC) name

15.1.4.1.229 Volume

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume
Property: volume (Optional)

Type: Object

Description: Information about the actual volume instance allocated to the workspace.

Examples:

- see https://kubernetes.io/docs/concepts/storage/persistent-volumes/

Table 319 – Properties for the volume object

Property	Type	Requirement	Description
uid	String	Optional	The unique identifier for the volume instance within its deployment context.
name	String	Optional	The name of the volume instance
mode	String	Optional	The mode for the volume instance.

Property	Type	Requirement	Description
path	String	Optional	The underlying path created from the actual volume.
sizeAllocated	String	Optional	The allocated size of the volume accessible to the associated workspace. This should include the scalar size as well as IEC standard unit in either decimal or binary form.
persistent	Boolean	Optional	Indicates if the volume persists beyond the life of the resource it is associated with.
remote	Boolean	Optional	Indicates if the volume is remotely (i.e., network) attached.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.230 Unique Identifier (UID)

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume/uid
Property: uid (Optional)

Type: String

Description: The unique identifier for the volume instance within its deployment context.

15.1.4.1.231 Name

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume/name
Property: name (Optional)

Type: String

Description: The name of the volume instance

15.1.4.1.232 Mode

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume/mode
Property: mode (Optional)

Type: String

Default Value: filesystem

Description: The mode for the volume instance.

Enumeration: Must be one of:

- filesystem
- block

15.1.4.1.233 Path

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume/path
Property: path (Optional)

Type: String

Description: The underlying path created from the actual volume.

15.1.4.1.234 Size allocated

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume/sizeAllocated
Property: sizeAllocated (Optional)

Type: String

Description: The allocated size of the volume accessible to the associated workspace. This should include the scalar size as well as IEC standard unit in either decimal or binary form.

Examples:

- 10GB
- 2Ti
- 1Pi

15.1.4.1.235 Persistent

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume/persistent
Property: persistent (Optional)

Type: Boolean

Description: Indicates if the volume persists beyond the life of the resource it is associated with.

15.1.4.1.236 Remote

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume/remote
Property: remote (Optional)

Type: Boolean

Description: Indicates if the volume is remotely (i.e., network) attached.

15.1.4.1.237 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.238 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 320 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.239 Name

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.240 Value

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

15.1.4.1.241 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.242 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 321 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.243 Name

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.244 Value

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.4.1.245 Runtime topology

Location: /formulation/[]/workflows/[]/tasks/[]/runtimeTopology
Property: runtimeTopology (Optional)

Type: Array

Description: A graph of the component runtime topology for task's instance. Each item of this array must be a Dependency object.

15.1.4.1.246 Dependency

Location: /formulation/[]/workflows/[]/tasks/[]/runtimeTopology/[]

Type: Object

Description: Defines the direct dependencies of a component, service, or the components provided/implemented by a given component. Components or services that do not have their own dependencies must be declared as empty elements within the graph. Components or services that are not represented in the dependency graph may have unknown dependencies. It is recommended that implementations assume this to be opaque and not an indicator of an object being dependency-free. It is recommended to leverage compositions to indicate unknown dependency graphs.

Table 322 – Properties for the runtimeTopology object

Property	Type	Requirement	Description
ref	String	Required	References a component or service by its bom-ref attribute
dependsOn	Array	Optional	The bom-ref identifiers of the components or services that are dependencies of this dependency object.
provides	Array	Optional	The bom-ref identifiers of the components or services that define a given specification or standard, which are provided or implemented by this dependency object. For example, a cryptographic library which implements a cryptographic algorithm. A component which implements another component does not imply that the implementation is in use.

15.1.4.1.247 Reference

Location: /formulation/[]/workflows/[]/tasks/[]/runtimeTopology/[]/ref
Property: ref (Required)

Type: String

Description: References a component or service by its bom-ref attribute

15.1.4.1.248 Depends On

Location: /formulation/[]/workflows/[]/tasks/[]/runtimeTopology/[]/dependsOn

Property: dependsOn (Optional)

Type: array (of String)

Description: The bom-ref identifiers of the components or services that are dependencies of this dependency object. Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType. Each item of this array must be a string.

All items must be unique.

15.1.4.1.249 Provides

Location: /formulation/[]/workflows/[]/tasks/[]/runtimeTopology/[]/provides

Property: provides (Optional)

Type: array (of String)

Description: The bom-ref identifiers of the components or services that define a given specification or standard, which are provided or implemented by this dependency object. For example, a cryptographic library which implements a cryptographic algorithm. A component which implements another component does not imply that the implementation is in use. Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType. Each item of this array must be a string.

All items must be unique.

15.1.4.1.250 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/properties

Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.251 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 323 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.252 Name

Location: /formulation/[]/workflows/[]/tasks/[]/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.253 Value

Location: /formulation/[]/workflows/[]/tasks/[]/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.4.1.254 Task dependency graph

Location: /formulation/[]/workflows/[]/taskDependencies
Property: taskDependencies (Optional)

Type: Array

Description: The graph of dependencies between tasks within the workflow. Each item of this array must be a Dependency object.

15.1.4.1.255 Dependency

Location: /formulation/[]/workflows/[]/taskDependencies/[]

Type: Object

Description: Defines the direct dependencies of a component, service, or the components provided/implemented by a given component. Components or services that do not have their own dependencies must be declared as empty elements within the graph. Components or services that are not represented in the dependency graph may have unknown dependencies. It is recommended that implementations assume this to be opaque and not an indicator of an object being dependency-free. It is recommended to leverage compositions to indicate unknown dependency graphs.

Table 324 – Properties for the taskDependencies object

Property	Type	Requirement	Description
ref	String	Required	References a component or service by its bom-ref attribute
dependsOn	Array	Optional	The bom-ref identifiers of the components or services that are dependencies of this dependency object.
provides	Array	Optional	The bom-ref identifiers of the components or services that define a given specification or standard, which are provided or implemented by this dependency object. For example, a cryptographic library which implements a cryptographic algorithm. A component which implements another component does not imply that the implementation is in use.

15.1.4.1.256 Reference

Location: /formulation/[]/workflows/[]/taskDependencies/[]/ref

Property: ref (Required)

Type: String

Description: References a component or service by its bom-ref attribute

15.1.4.1.257 Depends On

Location: /formulation/[]/workflows/[]/taskDependencies/[]/dependsOn
Property: dependsOn (Optional)

Type: array (of String)

Description: The bom-ref identifiers of the components or services that are dependencies of this dependency object. Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType. Each item of this array must be a string.

All items must be unique.

15.1.4.1.258 Provides

Location: /formulation/[]/workflows/[]/taskDependencies/[]/provides
Property: provides (Optional)

Type: array (of String)

Description: The bom-ref identifiers of the components or services that define a given specification or standard, which are provided or implemented by this dependency object. For example, a cryptographic library which implements a cryptographic algorithm. A component which implements another component does not imply that the implementation is in use. Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType. Each item of this array must be a string.

All items must be unique.

15.1.4.1.259 Task types

Location: /formulation/[]/workflows/[]/taskTypes
Property: taskTypes (Required)

Type: array (of String)

Description: Indicates the types of activities performed by the set of workflow tasks. Each item of this array must be a string.

Table 325 – Enumeration of possible values

Value	Description
copy	A task that copies software or data used to accomplish other tasks in the workflow.
clone	A task that clones a software repository into the workflow in order to retrieve its source code or data for use in a build step.
lint	A task that checks source code for programmatic and stylistic errors.
scan	A task that performs a scan against source code, or built or deployed components and services. Scans are typically run to gather or test for security vulnerabilities or policy compliance.
merge	A task that merges changes or fixes into source code prior to a build step in the workflow.
build	A task that builds the source code, dependencies and/or data into an artefact that can be deployed to and executed on target systems.

Value	Description
test	A task that verifies the functionality of a component or service.
deliver	A task that delivers a built artefact to one or more target repositories or storage systems.
deploy	A task that deploys a built artefact for execution on one or more target systems.
release	A task that releases a built, versioned artefact to a target repository or distribution system.
clean	A task that cleans unnecessary tools, build artifacts and/or data from workflow storage.
other	A workflow task that does not match current task type definitions.

15.1.4.1.260 Trigger

Location: /formulation/[]/workflows/[]/trigger
Property: trigger (Optional)

Type: Object

Description: The trigger that initiated the task.

Table 326 – Properties for the trigger object

Property	Type	Requirement	Description
bom-ref	String	Required	Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
uid	String	Required	The unique identifier for the resource instance within its deployment context.
name	String	Optional	The name of the resource instance.
description	String	Optional	A description of the resource instance.
resourceReferences	Array	Optional	References to component or service resources that are used to realize the resource instance.
type	String	Required	The source type of event which caused the trigger to fire.
event	Object	Optional	The event data that caused the associated trigger to activate.
conditions	Array	Optional	A list of conditions used to determine if a trigger should be activated.
timeActivated	String	Optional	The date and time (timestamp) when the trigger was activated.
inputs	Array	Optional	Represents resources and data brought into a task at runtime by executor or task commands
outputs	Array	Optional	Represents resources and data output from a task at runtime by executor or task commands

Property	Type	Requirement	Description
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.261 BOM Reference

Location: /formulation/[]/workflows/[]/trigger/bom-ref
Property: bom-ref (Required)

Type: String

Description: Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

15.1.4.1.262 Unique Identifier (UID)

Location: /formulation/[]/workflows/[]/trigger/uid
Property: uid (Required)

Type: String

Description: The unique identifier for the resource instance within its deployment context.

15.1.4.1.263 Name

Location: /formulation/[]/workflows/[]/trigger/name
Property: name (Optional)

Type: String

Description: The name of the resource instance.

15.1.4.1.264 Description

Location: /formulation/[]/workflows/[]/trigger/description
Property: description (Optional)

Type: String

Description: A description of the resource instance.

15.1.4.1.265 Resource references

Location: /formulation/[]/workflows/[]/trigger/resourceReferences
Property: resourceReferences (Optional)

Type: Array

Description: References to component or service resources that are used to realize the resource instance. Each item of this array must be a Resource reference choice object.

15.1.4.1.266 Resource reference choice

Location: /formulation/[]/workflows/[]/trigger/resourceReferences/[]

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Table 327 – Properties for the resourceReferences object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.267 BOM Reference

Location: /formulation/[]/workflows/[]/trigger/resourceReferences/[]/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.268 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.269 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.270 External Reference

Location: /formulation/[]/workflows/[]/trigger/resourceReferences/[]/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

All items must be unique.

15.1.4.1.271 Type

Location: /formulation/[]/workflows/[]/trigger/type
Property: type (Required)

Type: String

Description: The source type of event which caused the trigger to fire.

Enumeration: Must be one of:

- manual
- api
- webhook
- scheduled

15.1.4.1.272 Event

Location: /formulation/[]/workflows/[]/trigger/event
Property: event (Optional)

Type: Object

Description: The event data that caused the associated trigger to activate.

Table 328 – Properties for the event object

Property	Type	Requirement	Description
uid	String	Optional	The unique identifier of the event.
description	String	Optional	A description of the event.
timeReceived	String	Optional	The date and time (timestamp) when the event was received.
data	Object	Optional	Specifies the metadata and content for an attachment.
source	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
target	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.273 Unique Identifier (UID)

Location: /formulation/[]/workflows/[]/trigger/event/uid
Property: uid (Optional)

Type: String

Description: The unique identifier of the event.

15.1.4.1.274 Description

Location: /formulation/[]/workflows/[]/trigger/event/description
Property: description (Optional)

Type: String

Description: A description of the event.

15.1.4.1.275 Time Received

Location: /formulation/[]/workflows/[]/trigger/event/timeReceived
Property: timeReceived (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time (timestamp) when the event was received.

15.1.4.1.276 Attachment

Location: /formulation/[]/workflows/[]/trigger/event/data
Property: data (Optional)

Type: Object

Description: Specifies the metadata and content for an attachment.

Table 329 – Properties for the data object

Property	Type	Requirement	Description
contentType	String	Optional	Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.
encoding	String	Optional	Specifies the optional encoding the text is represented in.
content	String	Required	The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

15.1.4.1.277 Content-Type

Location: /formulation/[]/workflows/[]/trigger/event/data/contentType
Property: contentType (Optional)

Type: String

Default Value: text/plain

Description: Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.

Examples:

- text/plain
- application/json
- image/png

15.1.4.1.278 Encoding

Location: /formulation/[]/workflows/[]/trigger/event/data/encoding
Property: encoding (Optional)

Type: String

Description: Specifies the optional encoding the text is represented in.

Table 330 – Enumeration of possible values

Value	Description
base64	Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string.

15.1.4.1.279 Attachment Text

Location: /formulation/[]/workflows/[]/trigger/event/data/content
Property: content (Required)

Type: String

Description: The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

15.1.4.1.280 Resource reference choice

Location: /formulation/[]/workflows/[]/trigger/event/source
Property: source (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Table 331 – Properties for the source object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.281 BOM Reference

Location: /formulation/[]/workflows/[]/trigger/event/source/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.282 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.283 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.284 External Reference

Location: /formulation/[]/workflows/[]/trigger/event/source/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.285 Resource reference choice

Location: /formulation/[]/workflows/[]/trigger/event/target
Property: target (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Table 332 – Properties for the target object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.286 BOM Reference

Location: /formulation/[]/workflows/[]/trigger/event/target/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.287 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.288 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.289 External Reference

Location: /formulation/[]/workflows/[]/trigger/event/target/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.290 Properties

Location: /formulation/[]/workflows/[]/trigger/event/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.291 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/trigger/event/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 333 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.292 Name

Location: /formulation/[]/workflows/[]/trigger/event/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.293 Value

Location: /formulation/[]/workflows/[]/trigger/event/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

15.1.4.1.294 Conditions

Location: /formulation/[]/workflows/[]/trigger/conditions
Property: conditions (Optional)

Type: Array

Description: A list of conditions used to determine if a trigger should be activated. Each item of this array must be a Condition object.

15.1.4.1.295 Condition

Location: /formulation/[]/workflows/[]/trigger/conditions/[]

Type: Object
Description: A condition that was used to determine a trigger should be activated.

Table 334 – Properties for the conditions object

Property	Type	Requirement	Description
description	String	Optional	Describes the set of conditions which cause the trigger to activate.
expression	String	Optional	The logical expression that was evaluated that determined the trigger should be fired.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.296 Description

Location: /formulation/[]/workflows/[]/trigger/conditions/[]/description
Property: description (Optional)

Type: String

Description: Describes the set of conditions which cause the trigger to activate.

15.1.4.1.297 Expression

Location: /formulation/[]/workflows/[]/trigger/conditions/[]/expression
Property: expression (Optional)

Type: String

Description: The logical expression that was evaluated that determined the trigger should be fired.

15.1.4.1.298 Properties

Location: /formulation/[]/workflows/[]/trigger/conditions/[]/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different

values. Property names of interest to the general public are encouraged to be registered in the [CycloneDX Property Taxonomy](). Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.299 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/trigger/conditions/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 335 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.300 Name

Location: /formulation/[]/workflows/[]/trigger/conditions/[]/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.301 Value

Location: /formulation/[]/workflows/[]/trigger/conditions/[]/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.4.1.302 Time activated

Location: /formulation/[]/workflows/[]/trigger/timeActivated
Property: timeActivated (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time (timestamp) when the trigger was activated.

15.1.4.1.303 Inputs

Location: /formulation/[]/workflows/[]/trigger/inputs
Property: inputs (Optional)

Type: Array

Description: Represents resources and data brought into a task at runtime by executor or task commands Each item of this array must be an Input type object.

Examples:

- a configurationfile which was declared as a localcomponentorexternalReference``

15.1.4.1.304 Input type

Location: /formulation/[]/workflows/[]/trigger/inputs/[]

Type: Object
Description: Type that represents various input data types and formats.

Table 336 – Properties for the inputs object

Property	Type	Requirement	Description
source	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
target	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
resource	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
parameters	Array	Optional	Inputs that have the form of parameters with names and values.
environmentVars	Array	Optional	Inputs that have the form of parameters with names and values.
data	Object	Optional	Specifies the metadata and content for an attachment.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.305 Resource reference choice

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/source
Property: source (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Examples:

- source code repository
- database

Table 337 – Properties for the source object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute

Property	Type	Requirement	Description
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.306 BOM Reference

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/source/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.307 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.308 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.309 External Reference

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/source/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.310 Resource reference choice

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/target
Property: target (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Examples:

- workspace
- directory

Table 338 – Properties for the target object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.311 BOM Reference

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/target/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.312 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.313 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.314 External Reference

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/target/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.315 Resource reference choice

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/resource
Property: resource (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Examples:

- a reference to a configuration file in a repository (i.e., a bom-ref)
- a reference to a scanning service used in a task (i.e., a bom-ref)

Table 339 – Properties for the resource object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.316 BOM Reference

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/resource/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.317 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.318 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.319 External Reference

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/resource/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.320 Parameters

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/parameters
Property: parameters (Optional)

Type: Array

Description: Inputs that have the form of parameters with names and values. Each item of this array must be a Parameter object.

15.1.4.1.321 Parameter

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/parameters/[]

Type: Object
Description: A representation of a functional parameter.

Table 340 – Properties for the parameters object

Property	Type	Requirement	Description
name	String	Optional	The name of the parameter.
value	String	Optional	The value of the parameter.
dataType	String	Optional	The data type of the parameter.

15.1.4.1.322 Name

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/parameters/[]/name
Property: name (Optional)

Type: String

Description: The name of the parameter.

15.1.4.1.323 Value

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/parameters/[]/value
Property: value (Optional)

Type: String

Description: The value of the parameter.

15.1.4.1.324 Data type

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/parameters/[]/dataType
Property: dataType (Optional)

Type: String

Description: The data type of the parameter.

All items must be unique.

15.1.4.1.325 Environment variables

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/environmentVars
Property: environmentVars (Optional)

Type: Array

Description: Inputs that have the form of parameters with names and values.

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/environmentVars/[]

*Must be **one of**:*

1. Lightweight name-value pair

15.1.4.1.326 Lightweight name-value pair

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 341 – Properties for lightweight name-value pair

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

Type: String

15.1.4.1.327 Name

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/environmentVars/[]/name
Property: environmentVars (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.328 Value

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/environmentVars/[]/value
Property: environmentVars (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.4.1.329 Attachment

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/data
Property: data (Optional)

Type: Object

Description: Specifies the metadata and content for an attachment.

Table 342 – Properties for the data object

Property	Type	Requirement	Description
contentType	String	Optional	Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.
encoding	String	Optional	Specifies the optional encoding the text is represented in.
content	String	Required	The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

15.1.4.1.330 Content-Type

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/data/contentType
Property: contentType (Optional)

Type: String

Default Value: text/plain

Description: Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.

Examples:

- text/plain
- application/json
- image/png

15.1.4.1.331 Encoding

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/data/encoding
Property: encoding (Optional)

Type: String

Description: Specifies the optional encoding the text is represented in.

Table 343 – Enumeration of possible values

Value	Description
base64	Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string.

15.1.4.1.332 Attachment Text

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/data/content
Property: content (Required)

Type: String

Description: The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

15.1.4.1.333 Properties

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.334 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 344 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.335 Name

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.336 Value

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.4.1.337 Outputs

Location: /formulation/[]/workflows/[]/trigger/outputs
Property: outputs (Optional)

Type: Array

Description: Represents resources and data output from a task at runtime by executor or task commands

Examples:

- a log file or metrics data produced by the task

Location: /formulation/[]/workflows/[]/trigger/outputs/[]

Type: Object

Table 345 – Properties for the outputs object

Property	Type	Requirement	Description
type	String	Optional	Describes the type of data output.

Property	Type	Requirement	Description
source	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
target	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
resource	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
data	Object	Optional	Specifies the metadata and content for an attachment.
environmentVars	Array	Optional	Outputs that have the form of environment variables.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.338 Type

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/type
Property: type (Optional)

Type: String

Description: Describes the type of data output.

Enumeration: Must be one of:

- artifact
- attestation
- log
- evidence
- metrics
- other

15.1.4.1.339 Resource reference choice

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/source
Property: source (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Table 346 – Properties for the source object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute

Property	Type	Requirement	Description
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.340 BOM Reference

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/source/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.341 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.342 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.343 External Reference

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/source/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.344 Resource reference choice

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/target
Property: target (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Examples:

- a log file described as an externalReference within its target domain.

Table 347 – Properties for the target object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.345 BOM Reference

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/target/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.346 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.347 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.348 External Reference

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/target/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.349 Resource reference choice

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/resource
Property: resource (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Examples:

- configuration file
- source code

- scanning service

Table 348 – Properties for the resource object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.350 BOM Reference

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/resource/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

Must be ***any of****:*

1. Ref
2. BOM-Link Element

15.1.4.1.351 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.352 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.353 External Reference

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/resource/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.354 Attachment

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/data
Property: data (Optional)

Type: Object

Description: Specifies the metadata and content for an attachment.

Table 349 – Properties for the data object

Property	Type	Requirement	Description
contentType	String	Optional	Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.
encoding	String	Optional	Specifies the optional encoding the text is represented in.
content	String	Required	The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

15.1.4.1.355 Content-Type

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/data/contentType
Property: contentType (Optional)

Type: String

Default Value: text/plain

Description: Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.

Examples:

- text/plain
- application/json
- image/png

15.1.4.1.356 Encoding

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/data/encoding
Property: encoding (Optional)

Type: String

Description: Specifies the optional encoding the text is represented in.

Table 350 – Enumeration of possible values

Value	Description
base64	Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string.

15.1.4.1.357 Attachment Text

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/data/content
Property: content (Required)

Type: String

Description: The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

15.1.4.1.358 Environment variables

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/environmentVars
Property: environmentVars (Optional)

Type: Array

Description: Outputs that have the form of environment variables.

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/environmentVars/[]

*Must be **one of**:*

1. Lightweight name-value pair

15.1.4.1.359 Lightweight name-value pair

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 351 – Properties for lightweight name-value pair

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

Type: String

15.1.4.1.360 Name

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/environmentVars/[]/name
Property: environmentVars (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.361 Value

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/environmentVars/[]/value
Property: environmentVars (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.4.1.362 Properties

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.363 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 352 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.364 Name

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.365 Value

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.4.1.366 Properties

Location: /formulation/[]/workflows/[]/trigger/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.367 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/trigger/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 353 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.368 Name

Location: /formulation/[]/workflows/[]/trigger/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.369 Value

Location: /formulation/[]/workflows/[]/trigger/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

15.1.4.1.370 Steps

Location: /formulation/[]/workflows/[]/steps
Property: steps (Optional)

Type: Array

Description: The sequence of steps for the task.

Location: /formulation/[]/workflows/[]/steps/[]

Type: Object

Description: Executes specific commands or tools in order to accomplish its owning task as part of a sequence.

Table 354 – Properties for the steps object

Property	Type	Requirement	Description
name	String	Optional	A name for the step.
description	String	Optional	A description of the step.
commands	Array	Optional	Ordered list of commands or directives for the step

Property	Type	Requirement	Description
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.371 Name

Location: /formulation/[]/workflows/[]/steps/[]/name
Property: name (Optional)

Type: String

Description: A name for the step.

15.1.4.1.372 Description

Location: /formulation/[]/workflows/[]/steps/[]/description
Property: description (Optional)

Type: String

Description: A description of the step.

15.1.4.1.373 Commands

Location: /formulation/[]/workflows/[]/steps/[]/commands
Property: commands (Optional)

Type: Array

Description: Ordered list of commands or directives for the step

Location: /formulation/[]/workflows/[]/steps/[]/commands/[]

Type: Object

Table 355 – Properties for the commands object

Property	Type	Requirement	Description
executed	String	Optional	A text representation of the executed command.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.374 Executed

Location: /formulation/[]/workflows/[]/steps/[]/commands/[]/executed
Property: executed (Optional)

Type: String

Description: A text representation of the executed command.

15.1.4.1.375 Properties

Location: /formulation/[]/workflows/[]/steps/[]/commands/[]/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.376 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/steps/[]/commands/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 356 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.377 Name

Location: /formulation/[]/workflows/[]/steps/[]/commands/[]/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.378 Value

Location: /formulation/[]/workflows/[]/steps/[]/commands/[]/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

15.1.4.1.379 Properties

Location: /formulation/[]/workflows/[]/steps/[]/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX

Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.380 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/steps/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 357 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.381 Name

Location: /formulation/[]/workflows/[]/steps/[]/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.382 Value

Location: /formulation/[]/workflows/[]/steps/[]/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.4.1.383 Inputs

Location: /formulation/[]/workflows/[]/inputs
Property: inputs (Optional)

Type: Array

Description: Represents resources and data brought into a task at runtime by executor or task commands Each item of this array must be an Input type object.

Examples:

- a configurationfile which was declared as a localcomponentorexternalReference``

15.1.4.1.384 Input type

Location: /formulation/[]/workflows/[]/inputs/[]

Type: Object

Description: Type that represents various input data types and formats.

Table 358 – Properties for the inputs object

Property	Type	Requirement	Description
source	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
target	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
resource	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
parameters	Array	Optional	Inputs that have the form of parameters with names and values.
environmentVars	Array	Optional	Inputs that have the form of parameters with names and values.
data	Object	Optional	Specifies the metadata and content for an attachment.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.385 Resource reference choice

Location: /formulation/[]/workflows/[]/inputs/[]/source
Property: source (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Examples:

- source code repository
- database

Table 359 – Properties for the source object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.386 BOM Reference

Location: /formulation/[]/workflows/[]/inputs/[]/source/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.387 Ref

Type: String

Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.388 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$

Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.389 External Reference

Location: /formulation/[]/workflows/[]/inputs/[]/source/externalReference

Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.390 Resource reference choice

Location: /formulation/[]/workflows/[]/inputs/[]/target

Property: target (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Examples:

- workspace
- directory

Table 360 – Properties for the target object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.391 BOM Reference

Location: /formulation/[]/workflows/[]/inputs/[]/target/ref

Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.392 Ref

Type: String

Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.393 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$

Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.394 External Reference

Location: /formulation/[]/workflows/[]/inputs/[]/target/externalReference

Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.395 Resource reference choice

Location: /formulation/[]/workflows/[]/inputs/[]/resource

Property: resource (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Examples:

- a reference to a configuration file in a repository (i.e., a bom-ref)
- a reference to a scanning service used in a task (i.e., a bom-ref)

Table 361 – Properties for the resource object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.396 BOM Reference

Location: /formulation/[]/workflows/[]/inputs/[]/resource/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.397 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.398 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.399 External Reference

Location: /formulation/[]/workflows/[]/inputs/[]/resource/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.400 Parameters

Location: /formulation/[]/workflows/[]/inputs/[]/parameters
Property: parameters (Optional)

Type: Array

Description: Inputs that have the form of parameters with names and values. Each item of this array must be a Parameter object.

15.1.4.1.401 Parameter

Location: /formulation/[]/workflows/[]/inputs/[]/parameters/[]

Type: Object
Description: A representation of a functional parameter.

Table 362 – Properties for the parameters object

Property	Type	Requirement	Description
name	String	Optional	The name of the parameter.
value	String	Optional	The value of the parameter.
dataType	String	Optional	The data type of the parameter.

15.1.4.1.402 Name

Location: /formulation/[]/workflows/[]/inputs/[]/parameters/[]/name
Property: name (Optional)

Type: String

Description: The name of the parameter.

15.1.4.1.403 Value

Location: /formulation/[]/workflows/[]/inputs/[]/parameters/[]/value
Property: value (Optional)

Type: String

Description: The value of the parameter.

15.1.4.1.404 Data type

Location: /formulation/[]/workflows/[]/inputs/[]/parameters/[]/dataType
Property: dataType (Optional)

Type: String

Description: The data type of the parameter.

All items must be unique.

15.1.4.1.405 Environment variables

Location: /formulation/[]/workflows/[]/inputs/[]/environmentVars
Property: environmentVars (Optional)

Type: Array

Description: Inputs that have the form of parameters with names and values.

Location: /formulation/[]/workflows/[]/inputs/[]/environmentVars/[]

*Must be **one of**:*

1. Lightweight name-value pair

15.1.4.1.406 Lightweight name-value pair

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 363 – Properties for lightweight name-value pair

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

Type: String

15.1.4.1.407 Name

Location: /formulation/[]/workflows/[]/inputs/[]/environmentVars/[]/name
Property: environmentVars (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.408 Value

Location: /formulation/[]/workflows/[]/inputs/[]/environmentVars/[]/value
Property: environmentVars (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.4.1.409 Attachment

Location: /formulation/[]/workflows/[]/inputs/[]/data
Property: data (Optional)

Type: Object

Description: Specifies the metadata and content for an attachment.

Table 364 – Properties for the data object

Property	Type	Requirement	Description
contentType	String	Optional	Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.
encoding	String	Optional	Specifies the optional encoding the text is represented in.
content	String	Required	The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

15.1.4.1.410 Content-Type

Location: /formulation/[]/workflows/[]/inputs/[]/data/contentType
Property: contentType (Optional)

Type: String

Default Value: text/plain

Description: Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.

Examples:

- text/plain
- application/json
- image/png

15.1.4.1.411 Encoding

Location: /formulation/[]/workflows/[]/inputs/[]/data/encoding
Property: encoding (Optional)

Type: String

Description: Specifies the optional encoding the text is represented in.

Table 365 – Enumeration of possible values

Value	Description
base64	Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string.

15.1.4.1.412 Attachment Text

Location: /formulation/[]/workflows/[]/inputs/[]/data/content
Property: content (Required)

Type: String

Description: The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

15.1.4.1.413 Properties

Location: /formulation/[]/workflows/[]/inputs/[]/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.414 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/inputs/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different

values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 366 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.415 Name

Location: /formulation/[]/workflows/[]/inputs/[]/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.416 Value

Location: /formulation/[]/workflows/[]/inputs/[]/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.4.1.417 Outputs

Location: /formulation/[]/workflows/[]/outputs
Property: outputs (Optional)

Type: Array

Description: Represents resources and data output from a task at runtime by executor or task commands

Examples:

- a log file or metrics data produced by the task

Location: /formulation/[]/workflows/[]/outputs/[]

Type: Object

Table 367 – Properties for the outputs object

Property	Type	Requirement	Description
type	String	Optional	Describes the type of data output.
source	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
target	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Property	Type	Requirement	Description
resource	Array	Optional	A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.
data	Object	Optional	Specifies the metadata and content for an attachment.
environmentVars	Array	Optional	Outputs that have the form of environment variables.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.418 Type

Location: /formulation/[]/workflows/[]/outputs/[]/type
Property: type (Optional)

Type: String

Description: Describes the type of data output.

Enumeration: Must be one of:

- artifact
- attestation
- log
- evidence
- metrics
- other

15.1.4.1.419 Resource reference choice

Location: /formulation/[]/workflows/[]/outputs/[]/source
Property: source (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Table 368 – Properties for the source object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.420 BOM Reference

Location: /formulation/[]/workflows/[]/outputs/[]/source/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.421 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.422 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.423 External Reference

Location: /formulation/[]/workflows/[]/outputs/[]/source/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.424 Resource reference choice

Location: /formulation/[]/workflows/[]/outputs/[]/target
Property: target (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Examples:

- a log file described as an externalReference within its target domain.

Table 369 – Properties for the target object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.425 BOM Reference

Location: /formulation/[]/workflows/[]/outputs/[]/target/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.426 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.427 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.428 External Reference

Location: /formulation/[]/workflows/[]/outputs/[]/target/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.429 Resource reference choice

Location: /formulation/[]/workflows/[]/outputs/[]/resource
Property: resource (Optional)

Type: Object

Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Examples:

- configuration file
- source code
- scanning service

Table 370 – Properties for the resource object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.430 BOM Reference

Location: /formulation/[]/workflows/[]/outputs/[]/resource/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.431 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.432 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.433 External Reference

Location: /formulation/[]/workflows/[]/outputs/[]/resource/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

15.1.4.1.434 Attachment

Location: /formulation/[]/workflows/[]/outputs/[]/data
Property: data (Optional)

Type: Object

Description: Specifies the metadata and content for an attachment.

Table 371 – Properties for the data object

Property	Type	Requirement	Description
contentType	String	Optional	Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.
encoding	String	Optional	Specifies the optional encoding the text is represented in.
content	String	Required	The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

15.1.4.1.435 Content-Type

Location: /formulation/[]/workflows/[]/outputs/[]/data/contentType
Property: contentType (Optional)

Type: String

Default Value: text/plain

Description: Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.

Examples:

- text/plain
- application/json
- image/png

15.1.4.1.436 Encoding

Location: /formulation/[]/workflows/[]/outputs/[]/data/encoding
Property: encoding (Optional)

Type: String

Description: Specifies the optional encoding the text is represented in.

Table 372 – Enumeration of possible values

Value	Description
base64	Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string.

15.1.4.1.437 Attachment Text

Location: /formulation/[]/workflows/[]/outputs/[]/data/content
Property: content (Required)

Type: String

Description: The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

15.1.4.1.438 Environment variables

Location: /formulation/[]/workflows/[]/outputs/[]/environmentVars
Property: environmentVars (Optional)

Type: Array

Description: Outputs that have the form of environment variables.

Location: /formulation/[]/workflows/[]/outputs/[]/environmentVars/[]

*Must be **one of**:*

1. Lightweight name-value pair

15.1.4.1.439 Lightweight name-value pair

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 373 – Properties for lightweight name-value pair

Property	**Type**	**Requirement**	**Description**
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

Type: String

15.1.4.1.440 Name

Location: /formulation/[]/workflows/[]/outputs/[]/environmentVars/[]/name
Property: environmentVars (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.441 Value

Location: /formulation/[]/workflows/[]/outputs/[]/environmentVars/[]/value
Property: environmentVars (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.4.1.442 Properties

Location: /formulation/[]/workflows/[]/outputs/[]/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.443 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/outputs/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 374 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.444 Name

Location: /formulation/[]/workflows/[]/outputs/[]/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.445 Value

Location: /formulation/[]/workflows/[]/outputs/[]/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.4.1.446 Time start

Location: /formulation/[]/workflows/[]/timeStart
Property: timeStart (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time (timestamp) when the task started.

15.1.4.1.447 Time end

Location: /formulation/[]/workflows/[]/timeEnd
Property: timeEnd (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time (timestamp) when the task ended.

15.1.4.1.448 Workspaces

Location: /formulation/[]/workflows/[]/workspaces
Property: workspaces (Optional)

Type: Array

Description: A set of named filesystem or data resource shareable by workflow tasks. Each item of this array must be a Workspace object.

15.1.4.1.449 Workspace

Location: /formulation/[]/workflows/[]/workspaces/[]

Type: Object

Description: A named filesystem or data resource shareable by workflow tasks.

Table 375 – Properties for the workspaces object

Property	Type	Requirement	Description
bom-ref	String	Required	Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
uid	String	Required	The unique identifier for the resource instance within its deployment context.
name	String	Optional	The name of the resource instance.
aliases	Array	Optional	The names for the workspace as referenced by other workflow tasks. Effectively, a name mapping so other tasks can use their own local name in their steps.
description	String	Optional	A description of the resource instance.
resourceReferences	Array	Optional	References to component or service resources that are used to realize the resource instance.
accessMode	String	Optional	Describes the read-write access control for the workspace relative to the owning resource instance.
mountPath	String	Optional	A path to a location on disk where the workspace will be available to the associated task's steps.
managedDataType	String	Optional	The name of a domain-specific data type the workspace represents.
volumeRequest	String	Optional	Identifies the reference to the request for a specific volume type and parameters.
volume	Object	Optional	Information about the actual volume instance allocated to the workspace.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the

Property	Type	Requirement	Description
			standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.450 BOM Reference

Location: /formulation/[]/workflows/[]/workspaces/[]/bom-ref
Property: bom-ref (Required)

Type: String

Description: Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

15.1.4.1.451 Unique Identifier (UID)

Location: /formulation/[]/workflows/[]/workspaces/[]/uid
Property: uid (Required)

Type: String

Description: The unique identifier for the resource instance within its deployment context.

15.1.4.1.452 Name

Location: /formulation/[]/workflows/[]/workspaces/[]/name
Property: name (Optional)

Type: String

Description: The name of the resource instance.

15.1.4.1.453 Aliases

Location: /formulation/[]/workflows/[]/workspaces/[]/aliases
Property: aliases (Optional)

Type: array (of String)

Description: The names for the workspace as referenced by other workflow tasks. Effectively, a name mapping so other tasks can use their own local name in their steps. Each item of this array must be a string.

15.1.4.1.454 Description

Location: /formulation/[]/workflows/[]/workspaces/[]/description
Property: description (Optional)

Type: String

Description: A description of the resource instance.

15.1.4.1.455 Resource references

Location: /formulation/[]/workflows/[]/workspaces/[]/resourceReferences
Property: resourceReferences (Optional)

Type: Array

Description: References to component or service resources that are used to realize the resource instance. Each item of this array must be a Resource reference choice object.

15.1.4.1.456 Resource reference choice

Location: /formulation/[]/workflows/[]/workspaces/[]/resourceReferences/[]

Type: Object
Description: A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Table 376 – Properties for the resourceReferences object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

15.1.4.1.457 BOM Reference

Location: /formulation/[]/workflows/[]/workspaces/[]/resourceReferences/[]/ref
Property: ref (Optional)

Description: References an object by its bom-ref attribute

*Must be **any of**:*

1. Ref
2. BOM-Link Element

15.1.4.1.458 Ref

Type: String
Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

15.1.4.1.459 BOM-Link Element

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Pattern Constraint: ^urn:cdx:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}/[1-9][0-9]*#.+$
Description: Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/

15.1.4.1.460 External Reference

Location: /formulation/[]/workflows/[]/workspaces/[]/resourceReferences/[]/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

All items must be unique.

15.1.4.1.461 Access mode

Location: /formulation/[]/workflows/[]/workspaces/[]/accessMode

Property: accessMode (Optional)

Type: String

Description: Describes the read-write access control for the workspace relative to the owning resource instance.

Enumeration: Must be one of:

- read-only
- read-write
- read-write-once
- write-once
- write-only

15.1.4.1.462 Mount path

Location: /formulation/[]/workflows/[]/workspaces/[]/mountPath
Property: mountPath (Optional)

Type: String

Description: A path to a location on disc where the workspace will be available to the associated task's steps.

15.1.4.1.463 Managed data type

Location: /formulation/[]/workflows/[]/workspaces/[]/managedDataType
Property: managedDataType (Optional)

Type: String

Description: The name of a domain-specific data type the workspace represents.

Examples:

- ConfigMap
- Secret

15.1.4.1.464 Volume request

Location: /formulation/[]/workflows/[]/workspaces/[]/volumeRequest
Property: volumeRequest (Optional)

Type: String

Description: Identifies the reference to the request for a specific volume type and parameters.

Examples:

- a kubernetes Persistent Volume Claim (PVC) name

15.1.4.1.465 Volume

Location: /formulation/[]/workflows/[]/workspaces/[]/volume
Property: volume (Optional)

Type: Object

Description: Information about the actual volume instance allocated to the workspace.

Examples:

- see https://kubernetes.io/docs/concepts/storage/persistent-volumes/

Table 377 – Properties for the volume object

Property	Type	Requirement	Description
uid	String	Optional	The unique identifier for the volume instance within its deployment context.
name	String	Optional	The name of the volume instance
mode	String	Optional	The mode for the volume instance.
path	String	Optional	The underlying path created from the actual volume.
sizeAllocated	String	Optional	The allocated size of the volume accessible to the associated workspace. This should include the scalar size as well as IEC standard unit in either decimal or binary form.
persistent	Boolean	Optional	Indicates if the volume persists beyond the life of the resource it is associated with.
remote	Boolean	Optional	Indicates if the volume is remotely (i.e., network) attached.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

15.1.4.1.466 Unique Identifier (UID)

Location: /formulation/[]/workflows/[]/workspaces/[]/volume/uid
Property: uid (Optional)

Type: String

Description: The unique identifier for the volume instance within its deployment context.

15.1.4.1.467 Name

Location: /formulation/[]/workflows/[]/workspaces/[]/volume/name
Property: name (Optional)

Type: String

Description: The name of the volume instance

15.1.4.1.468 Mode

Location: /formulation/[]/workflows/[]/workspaces/[]/volume/mode
Property: mode (Optional)

Type: String

Default Value: filesystem

Description: The mode for the volume instance.

Enumeration: Must be one of:

- filesystem

- block

15.1.4.1.469 Path

Location: /formulation/[]/workflows/[]/workspaces/[]/volume/path
Property: path (Optional)

Type: String

Description: The underlying path created from the actual volume.

15.1.4.1.470 Size allocated

Location: /formulation/[]/workflows/[]/workspaces/[]/volume/sizeAllocated
Property: sizeAllocated (Optional)

Type: String

Description: The allocated size of the volume accessible to the associated workspace. This should include the scalar size as well as IEC standard unit in either decimal or binary form.

Examples:

- 10GB
- 2Ti
- 1Pi

15.1.4.1.471 Persistent

Location: /formulation/[]/workflows/[]/workspaces/[]/volume/persistent
Property: persistent (Optional)

Type: Boolean

Description: Indicates if the volume persists beyond the life of the resource it is associated with.

15.1.4.1.472 Remote

Location: /formulation/[]/workflows/[]/workspaces/[]/volume/remote
Property: remote (Optional)

Type: Boolean

Description: Indicates if the volume is remotely (i.e., network) attached.

15.1.4.1.473 Properties

Location: /formulation/[]/workflows/[]/workspaces/[]/volume/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.474 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/workspaces/[]/volume/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 378 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.475 Name

Location: /formulation/[]/workflows/[]/workspaces/[]/volume/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.476 Value

Location: /formulation/[]/workflows/[]/workspaces/[]/volume/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

15.1.4.1.477 Properties

Location: /formulation/[]/workflows/[]/workspaces/[]/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.478 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/workspaces/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 379 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.479 Name

Location: /formulation/[]/workflows/[]/workspaces/[]/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.480 Value

Location: /formulation/[]/workflows/[]/workspaces/[]/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.4.1.481 Runtime topology

Location: /formulation/[]/workflows/[]/runtimeTopology
Property: runtimeTopology (Optional)

Type: Array

Description: A graph of the component runtime topology for workflow's instance. Each item of this array must be a Dependency object.

15.1.4.1.482 Dependency

Location: /formulation/[]/workflows/[]/runtimeTopology/[]

Type: Object

Description: Defines the direct dependencies of a component, service, or the components provided/implemented by a given component. Components or services that do not have their own dependencies must be declared as empty elements within the graph. Components or services that are not represented in the dependency graph may have unknown dependencies. It is recommended that implementations assume this to be opaque and not an indicator of an object being dependency-free. It is recommended to leverage compositions to indicate unknown dependency graphs.

Table 380 – Properties for the runtimeTopology object

Property	Type	Requirement	Description
ref	String	Required	References a component or service by its bom-ref attribute
dependsOn	Array	Optional	The bom-ref identifiers of the components or services that are dependencies of this dependency object.
provides	Array	Optional	The bom-ref identifiers of the components or services that define a given specification or standard, which are provided or implemented by this

Property	Type	Requirement	Description
			dependency object. For example, a cryptographic library which implements a cryptographic algorithm. A component which implements another component does not imply that the implementation is in use.

15.1.4.1.483 Reference

Location: /formulation/[]/workflows/[]/runtimeTopology/[]/ref
Property: ref (Required)

Type: String

Description: References a component or service by its bom-ref attribute

15.1.4.1.484 Depends On

Location: /formulation/[]/workflows/[]/runtimeTopology/[]/dependsOn
Property: dependsOn (Optional)

Type: array (of String)

Description: The bom-ref identifiers of the components or services that are dependencies of this dependency object. Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType. Each item of this array must be a string.

All items must be unique.

15.1.4.1.485 Provides

Location: /formulation/[]/workflows/[]/runtimeTopology/[]/provides
Property: provides (Optional)

Type: array (of String)

Description: The bom-ref identifiers of the components or services that define a given specification or standard, which are provided or implemented by this dependency object. For example, a cryptographic library which implements a cryptographic algorithm. A component which implements another component does not imply that the implementation is in use. Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType. Each item of this array must be a string.

All items must be unique.

15.1.4.1.486 Properties

Location: /formulation/[]/workflows/[]/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.4.1.487 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 381 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.4.1.488 Name

Location: /formulation/[]/workflows/[]/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.4.1.489 Value

Location: /formulation/[]/workflows/[]/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

All items must be unique.

15.1.5 Properties

Location: /formulation/[]/properties
Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

15.1.5.1 Lightweight name-value pair

Location: /formulation/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 382 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

15.1.5.1.1 Name

Location: /formulation/[]/properties/[]/name
Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

15.1.5.1.2 Value

Location: /formulation/[]/properties/[]/value
Property: value (Optional)

Type: String

Description: The value of the property.

All items must be unique.

16 Declarations

Location: /declarations
Property: declarations (Optional)

Type: Object

Description: The list of declarations which describe the conformance to standards. Each declaration may include attestations, claims, and evidence.

Table 383 – Properties for the declarations object

Property	Type	Requirement	Description
assessors	Array	Optional	The list of assessors evaluating claims and determining conformance to requirements and confidence in that assessment.
attestations	Array	Optional	The list of attestations asserted by an assessor that maps requirements to claims.
claims	Array	Optional	The list of claims.
evidence	Array	Optional	The list of evidence
targets	Object	Optional	The list of targets which claims are made against.
affirmation	Object	Optional	A concise statement affirmed by an individual regarding all declarations, often used for third-party auditor acceptance or recipient acknowledgment. It includes a list of authorized signatories who assert the validity of the document on behalf of the organization.

Property	Type	Requirement	Description
signature	Array	Optional	Enveloped signature in JSON Signature Format (JSF).

16.1 Assessors

Location: /declarations/assessors
Property: assessors (Optional)

Type: Array

Description: The list of assessors evaluating claims and determining conformance to requirements and confidence in that assessment. Each item of this array must be an Assessor object.

16.1.1 Assessor

Location: /declarations/assessors/[]

Type: Object
Description: The assessor who evaluates claims and determines conformance to requirements and confidence in that assessment.

Table 384 – Properties for the assessors object

Property	Type	Requirement	Description
bom-ref	String	Optional	Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
thirdParty	Boolean	Optional	The boolean indicating if the assessor is outside the organization generating claims. A value of false indicates a self assessor.
organization	Object	Optional	The entity issuing the assessment.

16.1.1.1 BOM Reference

Location: /declarations/assessors/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.1.1.2 Third Party

Location: /declarations/assessors/[]/thirdParty
Property: thirdParty (Optional)

Type: Boolean

Description: The boolean indicating if the assessor is outside the organization generating claims. A value of false indicates a self assessor.

16.1.1.3 Organizational Entity

Location: /declarations/assessors/[]/organization
Property: organization (Optional)

Type: Object

Description: The entity issuing the assessment.

Table 385 – Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

16.1.1.3.1 BOM Reference

Location: /declarations/assessors/[]/organization/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.1.1.3.2 Organization Name

Location: /declarations/assessors/[]/organization/name
Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

16.1.1.3.3 Organization Address

Location: /declarations/assessors/[]/organization/address
Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 386 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

16.1.1.3.4 BOM Reference

Location: /declarations/assessors/[]/organization/address/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.1.1.3.5 Country

Location: /declarations/assessors/[]/organization/address/country
Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

16.1.1.3.6 Region

Location: /declarations/assessors/[]/organization/address/region
Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

16.1.1.3.7 Locality

Location: /declarations/assessors/[]/organization/address/locality
Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

16.1.1.3.8 Post Office Box Number

Location: /declarations/assessors/[]/organization/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

16.1.1.3.9 Postal Code

Location: /declarations/assessors/[]/organization/address/postalCode
Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

16.1.1.3.10 Street Address

Location: /declarations/assessors/[]/organization/address/streetAddress
Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

16.1.1.3.11 Organization URL(s)

Location: /declarations/assessors/[]/organization/url
Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

16.1.1.3.12 Organizational Contact

Location: /declarations/assessors/[]/organization/contact
Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

16.1.1.3.13 Organizational Contact

Location: /declarations/assessors/[]/organization/contact/[]

Type: Object

Table 387 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

16.1.1.3.14 BOM Reference

Location: /declarations/assessors/[]/organization/contact/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.1.1.3.15 Name

Location: /declarations/assessors/[]/organization/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

16.1.1.3.16 Email Address

Location: /declarations/assessors/[]/organization/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

16.1.1.3.17 Phone

Location: /declarations/assessors/[]/organization/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

16.2 Attestations

Location: /declarations/attestations
Property: attestations (Optional)

Type: Array

Description: The list of attestations asserted by an assessor that maps requirements to claims. Each item of this array must be an Attestation object.

16.2.1 Attestation

Location: /declarations/attestations/[]

Type: Object

Table 388 – Properties for the attestations object

Property	Type	Requirement	Description
summary	String	Optional	The short description explaining the main points of the attestation.
assessor	String	Optional	Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.
map	Array	Optional	The grouping of requirements to claims and the attestors declared conformance and confidence thereof.
signature	Array	Optional	Enveloped signature in JSON Signature Format (JSF).

16.2.1.1 Summary

Location: /declarations/attestations/[]/summary
Property: summary (Optional)

Type: String

Description: The short description explaining the main points of the attestation.

16.2.1.2 Assessor

Location: /declarations/attestations/[]/assessor
Property: assessor (Optional)

Type: String

Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

16.2.1.3 Map

Location: /declarations/attestations/[]/map
Property: map (Optional)

Type: Array

Description: The grouping of requirements to claims and the attestors declared conformance and confidence thereof. Each item of this array must be a Map object.

16.2.1.3.1 Map

Location: /declarations/attestations/[]/map/[]

Type: Object

Table 389 – Properties for the map object

Property	Type	Requirement	Description
requirement	String	Optional	Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.
claims	Array	Optional	The list of bom-ref to the claims being attested to.
counterClaims	Array	Optional	The list of bom-ref to the counter claims being attested to.
conformance	Object	Optional	The conformance of the claim meeting a requirement.
confidence	Object	Optional	The confidence of the claim meeting the requirement.

16.2.1.3.2 Requirement

Location: /declarations/attestations/[]/map/[]/requirement

Property: requirement (Optional)

Type: String

Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

16.2.1.3.3 Claims

Location: /declarations/attestations/[]/map/[]/claims

Property: claims (Optional)

Type: array (of String)

Description: The list of bom-ref to the claims being attested to. Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType. Each item of this array must be a string.

16.2.1.3.4 Counter Claims

Location: /declarations/attestations/[]/map/[]/counterClaims

Property: counterClaims (Optional)

Type: array (of String)

Description: The list of bom-ref to the counter claims being attested to. Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType. Each item of this array must be a string.

16.2.1.3.5 Conformance

Location: /declarations/attestations/[]/map/[]/conformance

Property: conformance (Optional)

Type: Object

Description: The conformance of the claim meeting a requirement.

Table 390 – Properties for the conformance object

Property	Type	Requirement	Description
score	Number	Optional	The conformance of the claim between and inclusive of 0 and 1, where 1 is 100% conformance.
rationale	String	Optional	The rationale for the conformance score.
mitigationStrategies	Array	Optional	The list of bom-ref to the evidence provided describing the mitigation strategies.

16.2.1.3.6 Score

Location: /declarations/attestations/[]/map/[]/conformance/score
Property: score (Optional)

Type: Number

Maximum Value: 1

Description: The conformance of the claim between and inclusive of 0 and 1, where 1 is 100% conformance.

16.2.1.3.7 Rationale

Location: /declarations/attestations/[]/map/[]/conformance/rationale
Property: rationale (Optional)

Type: String

Description: The rationale for the conformance score.

16.2.1.3.8 Mitigation Strategies

Location: /declarations/attestations/[]/map/[]/conformance/mitigationStrategies
Property: mitigationStrategies (Optional)

Type: array (of String)

Description: The list of bom-ref to the evidence provided describing the mitigation strategies. Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType. Each item of this array must be a string.

16.2.1.3.9 Confidence

Location: /declarations/attestations/[]/map/[]/confidence
Property: confidence (Optional)

Type: Object

Description: The confidence of the claim meeting the requirement.

Table 391 – Properties for the confidence object

Property	Type	Requirement	Description
score	Number	Optional	The confidence of the claim between and inclusive of 0 and 1, where 1 is 100% confidence.
rationale	String	Optional	The rationale for the confidence score.

16.2.1.3.10 Score

Location: /declarations/attestations/[]/map/[]/confidence/score
Property: score (Optional)

Type: Number

Maximum Value: 1

Description: The confidence of the claim between and inclusive of 0 and 1, where 1 is 100% confidence.

16.2.1.3.11 Rationale

Location: /declarations/attestations/[]/map/[]/confidence/rationale
Property: rationale (Optional)

Type: String

Description: The rationale for the confidence score.

16.2.1.4 Signature

Location: /declarations/attestations/[]/signature
Property: signature (Optional)

Type: Object

Description: An enveloped digital signature embedded within and specific to this object within the BOM. CycloneDX signatures enable integrity and authenticity verification without separating the signature from the BOM. Enveloped signatures enable each party in the supply chain to take responsibility for and sign their specific data, ensuring its integrity and authenticity. By aggregating all signatures, stakeholders can independently verify discrete pieces of information from each provider, enhancing overall transparency and trust in the supply chain.

Reference: Refer to the JSON Signature Format specification or to the XML Signature specification for implementation details.

- https://cyberphone.github.io/doc/security/jsf.html
- https://www.w3.org/TR/xmldsig-core/

16.3 Claims

Location: /declarations/claims
Property: claims (Optional)

Type: Array

Description: The list of claims. Each item of this array must be a Claim object.

16.3.1 Claim

Location: /declarations/claims/[]

Type: Object

Table 392 – Properties for the claims object

Property	Type	Requirement	Description
bom-ref	String	Optional	Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
target	String	Optional	Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.
predicate	String	Optional	The specific statement or assertion about the target.
mitigationStrategies	Array	Optional	The list of bom-ref to the evidence provided describing the mitigation strategies. Each mitigation strategy should include an explanation of how any weaknesses in the evidence will be mitigated.
reasoning	String	Optional	The written explanation of why the evidence provided substantiates the claim.
evidence	Array	Optional	The list of bom-ref to evidence that supports this claim.
counterEvidence	Array	Optional	The list of bom-ref to counterEvidence that supports this claim.
externalReferences	Array	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.
signature	Array	Optional	Enveloped signature in JSON Signature Format (JSF).

16.3.1.1 BOM Reference

Location: /declarations/claims/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.3.1.2 Target

Location: /declarations/claims/[]/target
Property: target (Optional)

Type: String

Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

16.3.1.3 Predicate

Location: /declarations/claims/[]/predicate
Property: predicate (Optional)

Type: String

Description: The specific statement or assertion about the target.

16.3.1.4 Mitigation Strategies

Location: /declarations/claims/[]/mitigationStrategies
Property: mitigationStrategies (Optional)

Type: array (of String)

Description: The list of bom-ref to the evidence provided describing the mitigation strategies. Each mitigation strategy should include an explanation of how any weaknesses in the evidence will be mitigated. Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType. Each item of this array must be a string.

16.3.1.5 Reasoning

Location: /declarations/claims/[]/reasoning
Property: reasoning (Optional)

Type: String

Description: The written explanation of why the evidence provided substantiates the claim.

16.3.1.6 Evidence

Location: /declarations/claims/[]/evidence
Property: evidence (Optional)

Type: array (of String)

Description: The list of bom-ref to evidence that supports this claim. Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType. Each item of this array must be a string.

16.3.1.7 Counter Evidence

Location: /declarations/claims/[]/counterEvidence
Property: counterEvidence (Optional)

Type: array (of String)

Description: The list of bom-ref to counterEvidence that supports this claim. Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType. Each item of this array must be a string.

16.3.1.8 External References

Location: /declarations/claims/[]/externalReferences
Property: externalReferences (Optional)

Type: Array

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM. Each item of this array must be an External Reference object.

Reference: Refer to the external reference definition at /externalReferences/[]

16.3.1.8.1 External Reference

Location: /declarations/claims/[]/externalReferences/[]

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

16.3.1.9 Signature

Location: /declarations/claims/[]/signature
Property: signature (Optional)

Type: Object

Description: An enveloped digital signature embedded within and specific to this object within the BOM. CycloneDX signatures enable integrity and authenticity verification without separating the signature from the BOM. Enveloped signatures enable each party in the supply chain to take responsibility for and sign their specific data, ensuring its integrity and authenticity. By aggregating all signatures, stakeholders can independently verify discrete pieces of information from each provider, enhancing overall transparency and trust in the supply chain.

Reference: Refer to the JSON Signature Format specification or to the XML Signature specification for implementation details.

- https://cyberphone.github.io/doc/security/jsf.html
- https://www.w3.org/TR/xmldsig-core/

16.4 Evidence

Location: /declarations/evidence
Property: evidence (Optional)

Type: Array

Description: The list of evidence Each item of this array must be an Evidence object.

16.4.1 Evidence

Location: /declarations/evidence/[]

Type: Object

Table 393 – Properties for the evidence object

Property	Type	Requirement	Description
bom-ref	String	Optional	Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
propertyName	String	Optional	The reference to the property name as defined in the CycloneDX Property Taxonomy.
description	String	Optional	The written description of what this evidence is and how it was created.
data	Array	Optional	The output or analysis that supports claims.
created	String	Optional	The date and time (timestamp) when the evidence was created.
expires	String	Optional	The optional date and time (timestamp) when the evidence is no longer valid.
author	Object	Optional	The author of the evidence.
reviewer	Object	Optional	The reviewer of the evidence.
signature	Array	Optional	Enveloped signature in JSON Signature Format (JSF).

16.4.1.1 BOM Reference

Location: /declarations/evidence/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.4.1.2 Property Name

Location: /declarations/evidence/[]/propertyName
Property: propertyName (Optional)

Type: String

Description: The reference to the property name as defined in the CycloneDX Property Taxonomy.

16.4.1.3 Description

Location: /declarations/evidence/[]/description
Property: description (Optional)

Type: String

Description: The written description of what this evidence is and how it was created.

16.4.1.4 Data

Location: /declarations/evidence/[]/data
Property: data (Optional)

Type: Array

Description: The output or analysis that supports claims. Each item of this array must be a Data object.

16.4.1.4.1 Data

Location: /declarations/evidence/[]/data/[]

Type: Object

Table 394 – Properties for the data object

Property	Type	Requirement	Description
name	String	Optional	The name of the data.
contents	Object	Optional	The contents or references to the contents of the data being described.
classification	String	Optional	Data classification tags data according to its type, sensitivity, and value if altered, stolen, or destroyed.
sensitiveData	Array	Optional	A description of any sensitive data included.
governance	Object	Optional	Data governance captures information regarding data ownership, stewardship, and custodianship, providing insights into the individuals or entities responsible for managing, overseeing, and safeguarding the data throughout its lifecycle.

16.4.1.4.2 Data Name

Location: /declarations/evidence/[]/data/[]/name
Property: name (Optional)

Type: String

Description: The name of the data.

16.4.1.4.3 Data Contents

Location: /declarations/evidence/[]/data/[]/contents
Property: contents (Optional)

Type: Object

Description: The contents or references to the contents of the data being described.

Table 395 – Properties for the contents object

Property	Type	Requirement	Description
attachment	Object	Optional	Specifies the metadata and content for an attachment.
url	String	Optional	The URL to where the data can be retrieved.

16.4.1.4.4 Attachment

Location: /declarations/evidence/[]/data/[]/contents/attachment
Property: attachment (Optional)

Type: Object

Description: Specifies the metadata and content for an attachment.

Table 396 – Properties for the attachment object

Property	Type	Requirement	Description
contentType	String	Optional	Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.
encoding	String	Optional	Specifies the optional encoding the text is represented in.
content	String	Required	The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

16.4.1.4.5 Content-Type

Location: /declarations/evidence/[]/data/[]/contents/attachment/contentType
Property: contentType (Optional)

Type: String

Default Value: text/plain

Description: Specifies the format and nature of the data being attached, helping systems correctly interpret and process the content. Common content type examples include application/json for JSON data and text/plain for plan text documents. RFC 2045 section 5.1 outlines the structure and use of content types. For a comprehensive list of registered content types, refer to the IANA media types registry.

Examples:

- text/plain
- application/json
- image/png

16.4.1.4.6 Encoding

Location: /declarations/evidence/[]/data/[]/contents/attachment/encoding
Property: encoding (Optional)

Type: String

Description: Specifies the optional encoding the text is represented in.

Table 397 – Enumeration of possible values

Value	Description
base64	Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string.

16.4.1.4.7 Attachment Text

Location: /declarations/evidence/[]/data/[]/contents/attachment/content
Property: content (Required)

Type: String

Description: The attachment data. Proactive controls such as input validation and sanitization should be employed to prevent misuse of attachment text.

16.4.1.4.8 Data URL

Location: /declarations/evidence/[]/data/[]/contents/url
Property: url (Optional)

Type: String

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL to where the data can be retrieved.

16.4.1.4.9 Data Classification

Location: /declarations/evidence/[]/data/[]/classification
Property: classification (Optional)

Type: String

Description: Data classification tags data according to its type, sensitivity, and value if altered, stolen, or destroyed.

16.4.1.4.10 Sensitive Data

Location: /declarations/evidence/[]/data/[]/sensitiveData
Property: sensitiveData (Optional)

Type: array (of String)

Description: A description of any sensitive data included. Each item of this array must be a string.

16.4.1.4.11 Data Governance

Location: /declarations/evidence/[]/data/[]/governance
Property: governance (Optional)

Type: Object

Description: Data governance captures information regarding data ownership, stewardship, and custodianship, providing insights into the individuals or entities responsible for managing, overseeing, and safeguarding the data throughout its lifecycle.

Table 398 – Properties for the governance object

Property	Type	Requirement	Description
custodians	Array	Optional	Data custodians are responsible for the safe custody, transport, and storage of data.
stewards	Array	Optional	Data stewards are responsible for data content, context, and associated business rules.
owners	Array	Optional	Data owners are concerned with risk and appropriate access to data.

16.4.1.4.12 Data Custodians

Location: /declarations/evidence/[]/data/[]/governance/custodians
Property: custodians (Optional)

Type: Array

Description: Data custodians are responsible for the safe custody, transport, and storage of data.

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]

Type: Object

Table 399 – Properties for the custodians object

Property	Type	Requirement	Description
organization	Object	Optional	The organization that is responsible for specific data governance role(s).
contact	Object	Optional	The individual that is responsible for specific data governance role(s).

16.4.1.4.13 Organization

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization
Property: organization (Optional)

Type: Object

Description: The organization that is responsible for specific data governance role(s).

Table 400 – Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

16.4.1.4.14 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.4.1.4.15 Organization Name

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/name
Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

16.4.1.4.16 Organization Address

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/address
Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 401 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

16.4.1.4.17 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/address/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.4.1.4.18 Country

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/address/country
Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

16.4.1.4.19 Region

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/address/region
Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

16.4.1.4.20 Locality

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/address/locality
Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

16.4.1.4.21 Post Office Box Number

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

16.4.1.4.22 Postal Code

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/address/postalCode
Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

16.4.1.4.23 Street Address

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/address/streetAddress
Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

16.4.1.4.24 Organization URL(s)

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/url
Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

16.4.1.4.25 Organizational Contact

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/contact
Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

16.4.1.4.26 Organizational Contact

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/contact/[]

Type: Object

Table 402 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

16.4.1.4.27 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/contact/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.4.1.4.28 Name

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

16.4.1.4.29 Email Address

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

16.4.1.4.30 Phone

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

16.4.1.4.31 Organizational Contact

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/contact
Property: contact (Optional)

Type: Object

Description: The individual that is responsible for specific data governance role(s).

Table 403 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

16.4.1.4.32 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/contact/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.4.1.4.33 Name

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/contact/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

16.4.1.4.34 Email Address

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/contact/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

16.4.1.4.35 Phone

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/contact/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

16.4.1.4.36 Data Stewards

Location: /declarations/evidence/[]/data/[]/governance/stewards
Property: stewards (Optional)

Type: Array

Description: Data stewards are responsible for data content, context, and associated business rules.

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]

Type: Object

Table 404 – Properties for the stewards object

Property	Type	Requirement	Description
organization	Object	Optional	The organization that is responsible for specific data governance role(s).
contact	Object	Optional	The individual that is responsible for specific data governance role(s).

16.4.1.4.37 Organization

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization
Property: organization (Optional)

Type: Object

Description: The organization that is responsible for specific data governance role(s).

Table 405 – Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

16.4.1.4.38 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.4.1.4.39 Organization Name

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/name
Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

16.4.1.4.40 Organization Address

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/address
Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 406 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

16.4.1.4.41 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/address/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.4.1.4.42 Country

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/address/country
Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

16.4.1.4.43 Region

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/address/region
Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

16.4.1.4.44 Locality

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/address/locality
Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

16.4.1.4.45 Post Office Box Number

Location:
/declarations/evidence/[]/data/[]/governance/stewards/[]/organization/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

16.4.1.4.46 Postal Code

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/address/postalCode
Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

16.4.1.4.47 Street Address

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/address/streetAddress
Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

16.4.1.4.48 Organization URL(s)

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/url
Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

16.4.1.4.49 Organizational Contact

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/contact
Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

16.4.1.4.50 Organizational Contact

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/contact/[]

Type: Object

Table 407 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

16.4.1.4.51 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/contact/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.4.1.4.52 Name

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

16.4.1.4.53 Email Address

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

16.4.1.4.54 Phone

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

16.4.1.4.55 Organizational Contact

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/contact
Property: contact (Optional)

Type: Object

Description: The individual that is responsible for specific data governance role(s).

Table 408 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

16.4.1.4.56 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/contact/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.4.1.4.57 Name

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/contact/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

16.4.1.4.58 Email Address

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/contact/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

16.4.1.4.59 Phone

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/contact/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

16.4.1.4.60 Data Owners

Location: /declarations/evidence/[]/data/[]/governance/owners
Property: owners (Optional)

Type: Array

Description: Data owners are concerned with risk and appropriate access to data.

Location: /declarations/evidence/[]/data/[]/governance/owners/[]

Type: Object

Table 409 – Properties for the owners object

Property	Type	Requirement	Description
organization	Object	Optional	The organization that is responsible for specific data governance role(s).
contact	Object	Optional	The individual that is responsible for specific data governance role(s).

16.4.1.4.61 Organization

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization
Property: organization (Optional)

Type: Object

Description: The organization that is responsible for specific data governance role(s).

Table 410 – Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

16.4.1.4.62 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.4.1.4.63 Organization Name

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/name

Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

16.4.1.4.64 Organization Address

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/address
Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 411 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

16.4.1.4.65 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/address/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.4.1.4.66 Country

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/address/country
Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

16.4.1.4.67 Region

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/address/region
Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

16.4.1.4.68 Locality

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/address/locality
Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

16.4.1.4.69 Post Office Box Number

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

16.4.1.4.70 Postal Code

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/address/postalCode
Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

16.4.1.4.71 Street Address

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/address/streetAddress
Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

16.4.1.4.72 Organization URL(s)

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/url
Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

16.4.1.4.73 Organizational Contact

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/contact
Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

16.4.1.4.74 Organizational Contact

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/contact/[]

Type: Object

Table 412 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

16.4.1.4.75 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/contact/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.4.1.4.76 Name

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

16.4.1.4.77 Email Address

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

16.4.1.4.78 Phone

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

16.4.1.4.79 Organizational Contact

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/contact
Property: contact (Optional)

Type: Object

Description: The individual that is responsible for specific data governance role(s).

Table 413 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

16.4.1.4.80 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/contact/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.4.1.4.81 Name

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/contact/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

16.4.1.4.82 Email Address

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/contact/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

16.4.1.4.83 Phone

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/contact/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

16.4.1.5 Created

Location: /declarations/evidence/[]/created
Property: created (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The date and time (timestamp) when the evidence was created.

16.4.1.6 Expires

Location: /declarations/evidence/[]/expires
Property: expires (Optional)

Type: String

Format: data-time as specified in [RFC 3339 section 5.6](https://www.ietf.org/rfc/rfc3339.html#section-5.6)

Description: The optional date and time (timestamp) when the evidence is no longer valid.

16.4.1.7 Organizational Contact

Location: /declarations/evidence/[]/author
Property: author (Optional)

Type: Object

Description: The author of the evidence.

Table 414 – Properties for the author object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

16.4.1.7.1 BOM Reference

Location: /declarations/evidence/[]/author/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.4.1.7.2 Name

Location: /declarations/evidence/[]/author/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

16.4.1.7.3 Email Address

Location: /declarations/evidence/[]/author/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

16.4.1.7.4 Phone

Location: /declarations/evidence/[]/author/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

16.4.1.8 Organizational Contact

Location: /declarations/evidence/[]/reviewer
Property: reviewer (Optional)

Type: Object

Description: The reviewer of the evidence.

Table 415 – Properties for the reviewer object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

16.4.1.8.1 BOM Reference

Location: /declarations/evidence/[]/reviewer/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.4.1.8.2 Name

Location: /declarations/evidence/[]/reviewer/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

16.4.1.8.3 Email Address

Location: /declarations/evidence/[]/reviewer/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

16.4.1.8.4 Phone

Location: /declarations/evidence/[]/reviewer/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

16.4.1.9 Signature

Location: /declarations/evidence/[]/signature
Property: signature (Optional)

Type: Object

Description: An enveloped digital signature embedded within and specific to this object within the BOM. CycloneDX signatures enable integrity and authenticity verification without separating the signature from the BOM. Enveloped signatures enable each party in the supply chain to take responsibility for and sign their specific data, ensuring its integrity and authenticity. By aggregating all signatures, stakeholders can independently verify discrete pieces of information from each provider, enhancing overall transparency and trust in the supply chain.

Reference: Refer to the JSON Signature Format specification or to the XML Signature specification for implementation details.

- https://cyberphone.github.io/doc/security/jsf.html
- https://www.w3.org/TR/xmldsig-core/

16.5 Targets

Location: /declarations/targets
Property: targets (Optional)

Type: Object

Description: The list of targets which claims are made against.

Table 416 – Properties for the targets object

Property	Type	Requirement	Description
organizations	Array	Optional	The list of organizations which claims are made against.
components	Array	Optional	The list of components which claims are made against.
services	Array	Optional	The list of services which claims are made against.

16.5.1 Organizations

Location: /declarations/targets/organizations
Property: organizations (Optional)

Type: Array

Description: The list of organizations which claims are made against. Each item of this array must be an Organizational Entity object.

16.5.1.1 Organizational Entity

Location: /declarations/targets/organizations/[]

Type: Object

Table 417 – Properties for the organizations object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

16.5.1.1.1 BOM Reference

Location: /declarations/targets/organizations/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.5.1.1.2 Organization Name

Location: /declarations/targets/organizations/[]/name
Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

16.5.1.1.3 Organization Address

Location: /declarations/targets/organizations/[]/address
Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 418 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

16.5.1.1.4 BOM Reference

Location: /declarations/targets/organizations/[]/address/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.5.1.1.5 Country

Location: /declarations/targets/organizations/[]/address/country
Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

16.5.1.1.6 Region

Location: /declarations/targets/organizations/[]/address/region
Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

16.5.1.1.7 Locality

Location: /declarations/targets/organizations/[]/address/locality
Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

16.5.1.1.8 Post Office Box Number

Location: /declarations/targets/organizations/[]/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

16.5.1.1.9 Postal Code

Location: /declarations/targets/organizations/[]/address/postalCode
Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

16.5.1.1.10 Street Address

Location: /declarations/targets/organizations/[]/address/streetAddress
Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

16.5.1.1.11 Organization URL(s)

Location: /declarations/targets/organizations/[]/url
Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

16.5.1.1.12 Organizational Contact

Location: /declarations/targets/organizations/[]/contact
Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

16.5.1.1.13 Organizational Contact

Location: /declarations/targets/organizations/[]/contact/[]

Type: Object

Table 419 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

16.5.1.1.14 BOM Reference

Location: /declarations/targets/organizations/[]/contact/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.5.1.1.15 Name

Location: /declarations/targets/organizations/[]/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

16.5.1.1.16 Email Address

Location: /declarations/targets/organizations/[]/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

16.5.1.1.17 Phone

Location: /declarations/targets/organizations/[]/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

16.5.2 Components

Location: /declarations/targets/components
Property: components (Optional)

Type: Array

Description: The list of components which claims are made against. Each item of this array must be a Component object.

16.5.2.1 Component

Location: /declarations/targets/components/[]

Type: Object
Reference: Refer to the component definition at /components/[]

16.5.3 Services

Location: /declarations/targets/services
Property: services (Optional)

Type: Array

Description: The list of services which claims are made against. Each item of this array must be a Service object.

16.5.3.1 Service

Location: /declarations/targets/services/[]

Type: Object
Reference: Refer to the service definition at /services/[]

16.6 Affirmation

Location: /declarations/affirmation
Property: affirmation (Optional)

Type: Object

Description: A concise statement affirmed by an individual regarding all declarations, often used for third-party auditor acceptance or recipient acknowledgment. It includes a list of authorized signatories who assert the validity of the document on behalf of the organization.

Table 420 – Properties for the affirmation object

Property	Type	Requirement	Description
statement	String	Optional	The brief statement affirmed by an individual regarding all declarations. *- Notes This could be an affirmation of acceptance by a third-party auditor or receiving individual of a file.
signatories	Array	Optional	The list of signatories authorized on behalf of an organization to assert validity of this document.
signature	Array	Optional	Enveloped signature in JSON Signature Format (JSF).

16.6.1 Statement

Location: /declarations/affirmation/statement
Property: statement (Optional)

Type: String

Description: The brief statement affirmed by an individual regarding all declarations. *- Notes This could be an affirmation of acceptance by a third-party auditor or receiving individual of a file.

Examples:

- I certify, to the best of my knowledge, that all information is correct.

16.6.2 Signatories

Location: /declarations/affirmation/signatories
Property: signatories (Optional)

Type: Array

Description: The list of signatories authorized on behalf of an organization to assert validity of this document. Each item of this array must be a Signatory object.

16.6.2.1 Signatory

Location: /declarations/affirmation/signatories/[]

Type: Object

Table 421 – Properties for the signatories object

Property	Type	Requirement	Description
name	String	Optional	The signatory's name.
role	String	Optional	The signatory's role within an organization.
signature	Array	Optional	Enveloped signature in JSON Signature Format (JSF).
organization	Object	Optional	The signatory's organization.
externalReference	Object	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

16.6.2.1.1 Name

Location: /declarations/affirmation/signatories/[]/name
Property: name (Optional)

Type: String

Description: The signatory's name.

16.6.2.1.2 Role

Location: /declarations/affirmation/signatories/[]/role
Property: role (Optional)

Type: String

Description: The signatory's role within an organization.

16.6.2.1.3 Signature

Location: /declarations/affirmation/signatories/[]/signature
Property: signature (Optional)

Type: Object

Description: An enveloped digital signature embedded within and specific to this object within the BOM. CycloneDX signatures enable integrity and authenticity verification without separating the signature from the BOM. Enveloped signatures enable each party in the supply chain to take responsibility for and sign their specific data, ensuring its integrity and authenticity. By aggregating all signatures, stakeholders can independently verify discrete pieces of information from each provider, enhancing overall transparency and trust in the supply chain.

Reference: Refer to the JSON Signature Format specification or to the XML Signature specification for implementation details.

- https://cyberphone.github.io/doc/security/jsf.html
- https://www.w3.org/TR/xmldsig-core/

16.6.2.1.4 Organizational Entity

Location: /declarations/affirmation/signatories/[]/organization
Property: organization (Optional)

Type: Object

Description: The signatory's organization.

Table 422 – Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the organization
address	Object	Optional	The physical address (location) of the organization
url	Array	Optional	The URL of the organization. Multiple URLs are allowed.
contact	Array	Optional	A contact at the organization. Multiple contacts are allowed.

16.6.2.1.5 BOM Reference

Location: /declarations/affirmation/signatories/[]/organization/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.6.2.1.6 Organization Name

Location: /declarations/affirmation/signatories/[]/organization/name
Property: name (Optional)

Type: String

Description: The name of the organization

Examples:

- Example Inc.

16.6.2.1.7 Organization Address

Location: /declarations/affirmation/signatories/[]/organization/address
Property: address (Optional)

Type: Object

Description: The physical address (location) of the organization

Table 423 – Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
country	String	Optional	The country name or the two-letter ISO 3166-1 country code.
region	String	Optional	The region or state in the country.
locality	String	Optional	The locality or city within the country.
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

16.6.2.1.8 BOM Reference

Location: /declarations/affirmation/signatories/[]/organization/address/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the address elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.6.2.1.9 Country

Location: /declarations/affirmation/signatories/[]/organization/address/country
Property: country (Optional)

Type: String

Description: The country name or the two-letter ISO 3166-1 country code.

16.6.2.1.10 Region

Location: /declarations/affirmation/signatories/[]/organization/address/region
Property: region (Optional)

Type: String

Description: The region or state in the country.

Examples:

- Texas

16.6.2.1.11 Locality

Location: /declarations/affirmation/signatories/[]/organization/address/locality
Property: locality (Optional)

Type: String

Description: The locality or city within the country.

Examples:

- Austin

16.6.2.1.12 Post Office Box Number

Location: /declarations/affirmation/signatories/[]/organization/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)

Type: String

Description: The post office box number.

Examples:

- 901

16.6.2.1.13 Postal Code

Location: /declarations/affirmation/signatories/[]/organization/address/postalCode
Property: postalCode (Optional)

Type: String

Description: The postal code.

Examples:

- 78758

16.6.2.1.14 Street Address

Location: /declarations/affirmation/signatories/[]/organization/address/streetAddress
Property: streetAddress (Optional)

Type: String

Description: The street address.

Examples:

- 100 Main Street

16.6.2.1.15 Organization URL(s)

Location: /declarations/affirmation/signatories/[]/organization/url
Property: url (Optional)

Type: array (of String)

Format: iri-reference as specified in [RFC 3987](https://www.ietf.org/rfc/rfc3987.html)

Description: The URL of the organization. Multiple URLs are allowed. Each item of this array must be a string.

Examples:

- https://example.com

16.6.2.1.16 Organizational Contact

Location: /declarations/affirmation/signatories/[]/organization/contact
Property: contact (Optional)

Type: Array

Description: A contact at the organization. Multiple contacts are allowed. Each item of this array must be an Organizational Contact object.

16.6.2.1.17 Organizational Contact

Location: /declarations/affirmation/signatories/[]/organization/contact/[]

Type: Object

Table 424 – Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of a contact
email	String	Optional	The email address of the contact.
phone	String	Optional	The phone number of the contact.

16.6.2.1.18 BOM Reference

Location: /declarations/affirmation/signatories/[]/organization/contact/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: An optional identifier which can be used to reference the object elsewhere in the BOM. Every bom-ref must be unique within the BOM. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

16.6.2.1.19 Name

Location: /declarations/affirmation/signatories/[]/organization/contact/[]/name
Property: name (Optional)

Type: String

Description: The name of a contact

Examples:

- Contact name

16.6.2.1.20 Email Address

Location: /declarations/affirmation/signatories/[]/organization/contact/[]/email
Property: email (Optional)

Type: String

Format: idn-email address as specified in [RFC 6531](https://www.ietf.org/rfc/rfc6531.html)

Description: The email address of the contact.

Examples:

- firstname.lastname@example.com

16.6.2.1.21 Phone

Location: /declarations/affirmation/signatories/[]/organization/contact/[]/phone
Property: phone (Optional)

Type: String

Description: The phone number of the contact.

Examples:

- 800-555-1212

16.6.2.1.22 External Reference

Location: /declarations/affirmation/signatories/[]/externalReference
Property: externalReference (Optional)

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

Reference: Refer to the external reference definition at /externalReferences/[]

16.6.3 Signature

Location: /declarations/affirmation/signature
Property: signature (Optional)

Type: Object

Description: An enveloped digital signature embedded within and specific to this object within the BOM. CycloneDX signatures enable integrity and authenticity verification without separating the signature from the BOM. Enveloped signatures enable each party in the supply chain to take responsibility for and sign their specific data, ensuring its integrity and authenticity. By aggregating all signatures, stakeholders can independently verify discrete pieces of information from each provider, enhancing overall transparency and trust in the supply chain.

Reference: Refer to the JSON Signature Format specification or to the XML Signature specification for implementation details.

- https://cyberphone.github.io/doc/security/jsf.html
- https://www.w3.org/TR/xmldsig-core/

16.7 Signature

Location: /declarations/signature
Property: signature (Optional)

Type: Object

Description: An enveloped digital signature embedded within and specific to this object within the BOM. CycloneDX signatures enable integrity and authenticity verification without separating the signature from the BOM. Enveloped signatures enable each party in the supply chain to take responsibility for and sign their specific data, ensuring its integrity and authenticity. By aggregating all signatures, stakeholders can independently verify discrete pieces of information from each provider, enhancing overall transparency and trust in the supply chain.

Reference: Refer to the JSON Signature Format specification or to the XML Signature specification for implementation details.

- https://cyberphone.github.io/doc/security/jsf.html
- https://www.w3.org/TR/xmldsig-core/

17 Definitions

Location: /definitions
Property: definitions (Optional)

Type: Object

Description: A collection of reusable objects that are defined and may be used elsewhere in the BOM.

Table 425 – Properties for the definitions object

Property	Type	Requirement	Description
standards	Array	Optional	The list of standards which may consist of regulations, industry or organizational-specific standards, maturity models, best practices, or any other requirements which can be evaluated against or attested to.

17.1 Standards

Location: /definitions/standards
Property: standards (Optional)

Type: Array

Description: The list of standards which may consist of regulations, industry or organizational-specific standards, maturity models, best practises, or any other requirements which can be evaluated against or attested to. Each item of this array must be a Standard object.

17.1.1 Standard

Location: /definitions/standards/[]

Type: Object

Description: A standard may consist of regulations, industry or organizational-specific standards, maturity models, best practises, or any other requirements which can be evaluated against or attested to.

Table 426 – Properties for the standards object

Property	Type	Requirement	Description
bom-ref	String	Optional	Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
name	String	Optional	The name of the standard. This will often be a shortened, single name of the standard.
version	String	Optional	The version of the standard.
description	String	Optional	The description of the standard.
owner	String	Optional	The owner of the standard, often the entity responsible for its release.
requirements	Array	Optional	The list of requirements comprising the standard.

Property	Type	Requirement	Description
levels	Array	Optional	The list of levels associated with the standard. Some standards have different levels of compliance.
externalReferences	Array	Optional	External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.
signature	Array	Optional	Enveloped signature in JSON Signature Format (JSF).

17.1.1.1 BOM Reference

Location: /definitions/standards/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

17.1.1.2 Name

Location: /definitions/standards/[]/name
Property: name (Optional)

Type: String

Description: The name of the standard. This will often be a shortened, single name of the standard.

17.1.1.3 Version

Location: /definitions/standards/[]/version
Property: version (Optional)

Type: String

Description: The version of the standard.

17.1.1.4 Description

Location: /definitions/standards/[]/description
Property: description (Optional)

Type: String

Description: The description of the standard.

17.1.1.5 Owner

Location: /definitions/standards/[]/owner
Property: owner (Optional)

Type: String

Description: The owner of the standard, often the entity responsible for its release.

17.1.1.6 Requirements

Location: /definitions/standards/[]/requirements
Property: requirements (Optional)

Type: Array

Description: The list of requirements comprising the standard. Each item of this array must be a Requirement object.

17.1.1.6.1 Requirement

Location: /definitions/standards/[]/requirements/[]

Type: Object

Table 427 – Properties for the requirements object

Property	Type	Requirement	Description
bom-ref	String	Optional	Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
identifier	String	Optional	The unique identifier used in the standard to identify a specific requirement. This should match what is in the standard and should not be the requirements bom-ref.
title	String	Optional	The title of the requirement.
text	String	Optional	The textual content of the requirement.
descriptions	Array	Optional	The supplemental text that provides additional guidance or context to the requirement, but is not directly part of the requirement.
openCre	Array	Optional	The Common Requirements Enumeration (CRE) identifier(s). CRE is a structured and standardized framework for uniting security standards and guidelines. CRE links each section of a resource to a shared topic identifier (a Common Requirement). Through this shared topic link, all resources map to each other. Use of CRE promotes clear and unambiguous communication among stakeholders.
parent	String	Optional	Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.
properties	Array	Optional	Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.
externalReferences	Array	Optional	External references provide a way to document systems, sites, and information that may be relevant, but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

17.1.1.6.2 BOM Reference

Location: /definitions/standards/[]/requirements/[]/bom-ref
Property: bom-ref (Optional)

Type: String

Description: Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

17.1.1.6.3 Identifier

Location: /definitions/standards/[]/requirements/[]/identifier
Property: identifier (Optional)

Type: String

Description: The unique identifier used in the standard to identify a specific requirement. This should match what is in the standard and should not be the requirements bom-ref.

17.1.1.6.4 Title

Location: /definitions/standards/[]/requirements/[]/title
Property: title (Optional)

Type: String

Description: The title of the requirement.

17.1.1.6.5 Text

Location: /definitions/standards/[]/requirements/[]/text
Property: text (Optional)

Type: String

Description: The textual content of the requirement.

17.1.1.6.6 Descriptions

Location: /definitions/standards/[]/requirements/[]/descriptions
Property: descriptions (Optional)

Type: array (of String)

Description: The supplemental text that provides additional guidance or context to the requirement, but is not directly part of the requirement. Each item of this array must be a string.

17.1.1.6.7 OWASP OpenCRE Identifier(s)

Location: /definitions/standards/[]/requirements/[]/openCre
Property: openCre (Optional)

Type: array (of String)

Pattern Constraint: ^CRE:[0-9]+-[0-9]+$

Description: The Common Requirements Enumeration (CRE) identifier(s). CRE is a structured and standardized framework for uniting security standards and guidelines. CRE links each section of a resource to a shared topic identifier (a Common Requirement). Through this shared topic link, all resources map to each other. Use of CRE promotes clear and unambiguous communication among stakeholders. Each item of this array must be a string.

Examples:

- CRE:764-507

17.1.1.6.8 Parent BOM Reference

Location: /definitions/standards/[]/requirements/[]/parent
Property: parent (Optional)

Type: String

Description: Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType.

17.1.1.6.9 Properties

Location: /definitions/standards/[]/requirements/[]/properties

Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

17.1.1.6.10 Lightweight name-value pair

Location: /definitions/standards/[]/requirements/[]/properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 428 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

17.1.1.6.11 Name

Location: /definitions/standards/[]/requirements/[]/properties/[]/name

Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

17.1.1.6.12 Value

Location: /definitions/standards/[]/requirements/[]/properties/[]/value

Property: value (Optional)

Type: String

Description: The value of the property.

17.1.1.6.13 External References

Location: /definitions/standards/[]/requirements/[]/externalReferences

Property: externalReferences (Optional)

Type: Array

Description: External references provide a way to document systems, sites, and information that may be relevant, but are not included with the BOM. They may also establish specific relationships within or external to the BOM. Each item of this array must be an External Reference object.

Reference: Refer to the external reference definition at /externalReferences/[]

17.1.1.6.14 External Reference

Location: /definitions/standards/[]/requirements/[]/externalReferences/[]

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

17.1.1.7 Levels

Location: /definitions/standards/[]/levels

Property: levels (Optional)

Type: Array

Description: The list of levels associated with the standard. Some standards have different levels of compliance. Each item of this array must be a Level object.

17.1.1.7.1 Level

Location: /definitions/standards/[]/levels/[]

Type: Object

Table 429 – Properties for the levels object

Property	Type	Requirement	Description
bom-ref	String	Optional	Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
identifier	String	Optional	The identifier used in the standard to identify a specific level.
title	String	Optional	The title of the level.
description	String	Optional	The description of the level.
requirements	Array	Optional	The list of requirement bom-refs that comprise the level.

17.1.1.7.2 BOM Reference

Location: /definitions/standards/[]/levels/[]/bom-ref

Property: bom-ref (Optional)

Type: String

Description: Identifier for referable and therefore interlinkable elements. Value SHOULD not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

17.1.1.7.3 Identifier

Location: /definitions/standards/[]/levels/[]/identifier

Property: identifier (Optional)

Type: String

Description: The identifier used in the standard to identify a specific level.

17.1.1.7.4 Title

Location: /definitions/standards/[]/levels/[]/title
Property: title (Optional)

Type: String

Description: The title of the level.

17.1.1.7.5 Description

Location: /definitions/standards/[]/levels/[]/description
Property: description (Optional)

Type: String

Description: The description of the level.

17.1.1.7.6 Requirements

Location: /definitions/standards/[]/levels/[]/requirements
Property: requirements (Optional)

Type: array (of String)

Description: The list of requirement bom-refs that comprise the level. Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to bomLinkElementType. Each item of this array must be a string.

17.1.1.8 External References

Location: /definitions/standards/[]/externalReferences
Property: externalReferences (Optional)

Type: Array

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM. Each item of this array must be an External Reference object.

Reference: Refer to the external reference definition at /externalReferences/[]

17.1.1.8.1 External Reference

Location: /definitions/standards/[]/externalReferences/[]

Type: Object

Description: External references provide a way to document systems, sites, and information that may be relevant but are not included with the BOM. They may also establish specific relationships within or external to the BOM.

17.1.1.9 Signature

Location: /definitions/standards/[]/signature
Property: signature (Optional)

Type: Object

Description: An enveloped digital signature embedded within and specific to this object within the BOM. CycloneDX signatures enable integrity and authenticity verification without separating the signature from the BOM. Enveloped signatures enable each party in the supply chain to take responsibility for and sign their specific data, ensuring its integrity and authenticity. By aggregating all signatures, stakeholders can independently verify discrete pieces of information from each provider, enhancing overall transparency and trust in the supply chain.

Reference: Refer to the JSON Signature Format specification or to the XML Signature specification for implementation details.

- https://cyberphone.github.io/doc/security/jsf.html
- https://www.w3.org/TR/xmldsig-core/

18 Properties

Location: /properties

Property: properties (Optional)

Type: Array

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional. Each item of this array must be a Lightweight name-value pair object.

18.1 Lightweight name-value pair

Location: /properties/[]

Type: Object

Description: Provides the ability to document properties in a name-value store. This provides flexibility to include data not officially supported in the standard without having to use additional namespaces or create extensions. Unlike key-value stores, properties support duplicate names, each potentially having different values. Property names of interest to the general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

Table 430 – Properties for the properties object

Property	Type	Requirement	Description
name	String	Required	The name of the property. Duplicate names are allowed, each potentially having a different value.
value	String	Optional	The value of the property.

18.1.1 Name

Location: /properties/[]/name

Property: name (Required)

Type: String

Description: The name of the property. Duplicate names are allowed, each potentially having a different value.

18.1.2 Value

Location: /properties/[]/value

Property: value (Optional)

Type: String

Description: The value of the property.

19 Signature

Location: /signature

Property: signature (Optional)

Type: Object

Description: An enveloped digital signature embedded within and specific to this object within the BOM. CycloneDX signatures enable integrity and authenticity verification without separating the signature from the BOM. Enveloped signatures enable each party in the supply chain to take responsibility for and sign their specific data, ensuring its integrity and authenticity. By aggregating all signatures, stakeholders can independently verify discrete pieces of information from each provider, enhancing overall transparency and trust in the supply chain.

Reference: Refer to the JSON Signature Format specification or to the XML Signature specification for implementation details.

- https://cyberphone.github.io/doc/security/jsf.html
- https://www.w3.org/TR/xmldsig-core/