ecma INTERNATIONAL

Standard ECMA-424

2nd Edition / December 2025

CycloneDX Bill of materials specification

Standard

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



is the registered trademark of Ecma International



COPYRIGHT PROTECTED DOCUMENT

© Ecma International 2025

Contents

Page

1 Scope . . . 1
2 Conformance . . . 1
3 Normative references . . . 2
4 Terms and definitions . . . 2
5 Overview . . . 4
5.1 Purpose and benefits . . . 4
5.2 Design philosophy and guiding principles . . . 4
5.3 CycloneDX object model . . . 7
6 CycloneDX Bill of Materials Standard . . . 12
6.1 BOM Format . . . 14
6.2 CycloneDX Specification Version . . . 14
6.3 BOM Serial Number . . . 14
6.4 BOM Version . . . 15
6.5 BOM Metadata . . . 15
6.5.1 Timestamp . . . 16
6.5.2 Lifecycles . . . 16
6.5.3 Tools . . . 18
6.5.4 Tools . . . 18
6.5.5 Tools (legacy) . . . 19
6.5.6 Components . . . 19
6.5.7 Services . . . 19
6.5.8 Tools (legacy) . . . 20
6.5.9 BOM Manufacturer . . . 23
6.5.10 BOM Authors . . . 27
6.5.11 Component . . . 29
6.5.12 Component Manufacture (legacy) . . . 29
6.5.13 Supplier . . . 33
6.5.14 BOM License(s) . . . 38
6.5.15 Properties . . . 38
6.5.16 Distribution Constraints . . . 39
6.6 Components . . . 40
6.6.1 Component . . . 40
6.7 Services . . . 264
6.7.1 Service . . . 264
6.8 External References . . . 308
6.8.1 External Reference . . . 309
6.9 Dependencies . . . 314
6.9.1 Dependency . . . 314
6.10 Compositions . . . 315
6.10.1 Compositions . . . 316
6.11 Vulnerabilities . . . 318
6.11.1 Vulnerability . . . 319
6.12 Annotations . . . 346
6.12.1 Annotations . . . 346
6.13 Formulation . . . 354
6.13.1 Formula . . . 354
6.14 Declarations . . . 470
6.14.1 Assessors . . . 471
6.14.2 Attestations . . . 476
6.14.3 Claims . . . 479
6.14.4 Evidence . . . 481
6.14.5 Targets . . . 505
6.14.6 Affirmation . . . 510
6.14.7 Signature . . . 516
6.15 Definitions . . . 516
6.15.1 Standards . . . 517

© Ecma International 2025

6.15.2 Patents . . . 523
6.16 Citations . . . 549
6.16.1 Citation . . . 549
6.17 Properties . . . 552
6.17.1 Lightweight name-value pair . . . 552
6.18 Signature . . . 552
Bibliography . . . 553
Colophon . . . 555
Software License . . . 557

© Ecma International 2025

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 December 2025.

© Ecma International 2025

About this specification

The document at https://tc54.org/ecma424/ is the most accurate and up-to-date CycloneDX specification.

This document is available as a single page and as multiple pages.

Contributing to this specification

This specification is developed on GitHub with the help of the OWASP community. There are a number of ways to contribute to the development of this specification:

GitHub Repository: https://github.com/Ecma-TC54/ECMA-424
Issues: All Issues, File a New Issue
Pull Requests: All Pull Requests, Create a New Pull Request
Editors:

- Jan Kowalleck
- Steve Springett

Community:

- Chat: Slack Workspace

Refer to the colophon for more information on how this document is created.

© Ecma International 2025

COPYRIGHT NOTICE

© 2025 Ecma International

By obtaining and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

This document may be copied, published and distributed to others, and certain derivative works of it may be prepared, copied, published, and distributed, in whole or in part, provided that the above copyright notice and this Copyright License and Disclaimer are included on all such copies and derivative works. The only derivative works that are permissible under this Copyright License and Disclaimer are:

(i) works which incorporate all or portion of this document for the purpose of providing commentary or explanation (such as an annotated version of the document),

(ii) works which incorporate all or portion of this document for the purpose of incorporating features that provide accessibility,

(iii) translations of this document into languages other than English and into different formats and

(iv) works by making use of this specification in standard conformant products by implementing (e.g. by copy and paste wholly or partly) the functionality therein.

However, the content of this document itself may not be modified in any way, including by removing the copyright notice or references to Ecma International, except as required to translate it into languages other than English or into a different format.

The official version of an Ecma International document is the English language version on the Ecma International website. In the event of discrepancies between a translated version and the official version, the official version shall govern.

The limited permissions granted above are perpetual and will not be revoked by Ecma International or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and ECMA INTERNATIONAL DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

CycloneDX Bill of materials specification

1 Scope

This Standard defines the CycloneDX v1.7 Bill of Materials (BOM) specification, which defines a structured format for representing detailed inventory information of software and hardware components, services, dependencies, vulnerabilities, cryptographic artefacts, machine learning models, and other elements relevant to supply chain transparency and cybersecurity assurance.

This Standard specifies the syntax and semantics for:

- describing software and hardware components, services, dependencies, vulnerabilities, and compositions;
- expressing metadata, annotations, external references, lifecycle context, and formulation processes;
- supporting domain-specific modelling for cryptographic artefacts and machine learning models;
- asserting claims, attestations, and supporting evidence for conformance to standards or requirements;
- documenting open-source and commercial licensing and other artefacts supporting software transparency and risk analysis.

The BOM is serialised using a machine-readable JSON format and is intended for exchange across tools, systems, and stakeholders within software and hardware supply chains.

NOTE 1 BOMs conforming to this Standard may be produced manually or generated by automated tools during any phase of the system or software lifecycle.

NOTE 2 This Standard does not define enforcement mechanisms for verifying the accuracy or completeness of a BOM, nor does it prescribe a specific transport mechanism for exchanging BOMs.

2 Conformance

This Standard includes the implementation requirements that systems processing CycloneDX content shall 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 shall 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 shall 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 shall 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 shall conform to this Standard.
- It shall 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 shall create or modify that feature in a manner conforming to this Standard.

© Ecma International 2025

3 Normative references

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

ECMA-404, *The JSON data interchange syntax*
https://ecma-international.org/publications-and-standards/standards/ecma-404/

IETF RFC 3339, *Date and Time on the Internet: Timestamps*
https://tools.ietf.org/html/rfc3339

IETF RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*
https://datatracker.ietf.org/doc/html/rfc3986

IETF RFC 3987, *Internationalized Resource Identifiers (IRIs)*
https://tools.ietf.org/html/rfc3987

IETF RFC 4122, *A Universally Unique IDentifier (UUID) URN Namespace*
https://datatracker.ietf.org/doc/html/rfc4122

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 document, the following terms and definitions apply. Terms explicitly defined in this Standard are not to be presumed to refer implicitly to similar terms defined elsewhere.

4.1
attestation
formal declaration that something is true or accurate

NOTE Attestations are often backed by documentation or verification from an authoritative source. An attestation serves as a confirmation or proof of a fact, condition, or compliance with specific standards or requirements.

4.2
author
person who creates written works, such as software or data

4.3
component function
purpose for which a software component exists

NOTE Examples of component functions include parsers, database persistence, and authentication providers.

4.4
component type
general classification of a software components architecture

© Ecma International 2025

NOTE Examples of component types include libraries, frameworks, applications, containers, and operating systems.

4.5
manufacturer
entity that develops and produces products such as virtual or physical goods

4.6
direct dependency
component that is referenced by a main (metadata) component itself

4.7
Package-URL (PURL)
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
process of agreeing to terms and acquiring physical or virtual goods or services

4.10
provenance
chain of custody and origin of a software component

NOTE Provenance incorporates the point of origin through distribution as well as derivatives in the case of software that has been modified.

4.11
provider
entity that offers services, infrastructure, or platforms

NOTE These services can include computing resources, storage, software applications, and networking capabilities.

4.12
publisher
entity that produces and distributes content, such as software, to the public

4.13
Software identification (SWID)
ISO standard that formalizes XML records that uniquely identify software products, versions, and installations to support asset management, security, and compliance

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

© Ecma International 2025

4.15
supplier
entity that provides products or services to another entity, typically within a supply chain

4.16
third-party component
software component not directly created

NOTE Third-party components may include open source, "source available", and commercial or proprietary software.

4.17
transitive dependency
software component that is indirectly used by another component by means of being a dependency of a dependency

5 Overview

This section contains a non-normative overview of CycloneDX.

CycloneDX is a modern standard for the software and system supply chain. Originating in the OWASP community in 2017, it has grown into a general-purpose Bill of Materials (BOM) standard capable of representing software, hardware, services, and other forms of inventory.

Backed by a formal governance model and global community support, CycloneDX has matured into an OWASP flagship project and, in December 2025, was formally adopted as an Ecma International Standard.

5.1 Purpose and benefits

At its core, CycloneDX enables software and system transparency. It provides detailed information about components such as versions, suppliers, and dependencies, allowing organizations to:

- Identify and manage risks across complex supply chains
- Facilitate compliance with regulatory and contractual requirements
- Build trust across the supply chain

This transparency supports a wide range of use cases.

CycloneDX continues to evolve to address emerging industry challenges. With each new release, it introduces features and improvements designed to stay ahead of the changing threat landscape and meet the needs of a broad spectrum of stakeholders, from developers and security teams to regulators and supply chain partners.

5.2 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

© Ecma International 2025

- Provide a smooth path to specification compliance through prescriptive design

Defining Software Bill of Materials

The U.S. Cybersecurity & Infrastructure Security Agency (CISA) defines software bill of 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.

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.

Conformance and Alignment

The CycloneDX Standard exceeds the requirements outlined in the CISA Minimum Elements for an SBOM as well as the SBOM requirements defined by the German Federal Office for Information Security (Bundesamt für Sicherheit in der Informationstechnik, BSI). In addition, CycloneDX supports and advances the secure development practices recommended in NIST's Secure Software Development Framework (SSDF), and provides the structured transparency needed to help organizations meet obligations under the European Union Cyber Resilience Act (CRA).

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

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

© Ecma International 2025

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.

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.

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.

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.

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.

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.

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.

© Ecma International 2025

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.

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.

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.

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."

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.

5.3 CycloneDX object model

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

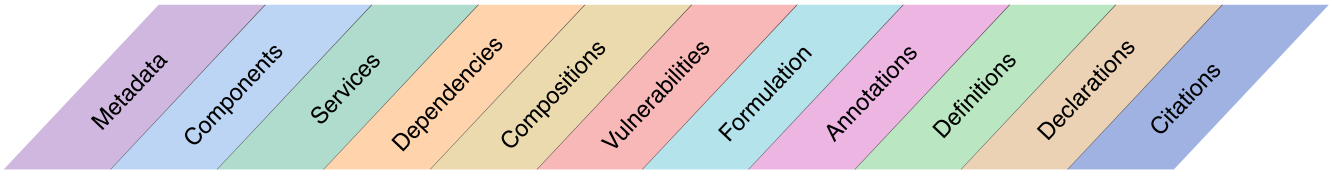


Figure 1 — Root Object Model Overview

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

© Ecma International 2025



Figure 2 — Object Model Swimlane Overview

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.

© Ecma International 2025

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 shall conform to RFC 4122. The use of serial numbers is recommended.

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'.

The Anatomy of a CycloneDX BOM

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

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.

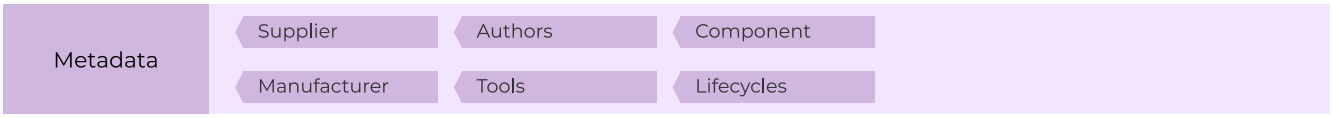


Figure 3 — Metadata Object Model Overview

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.

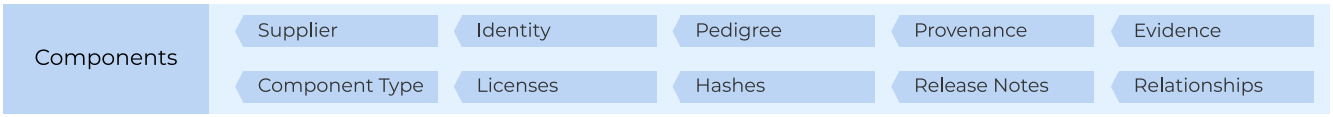


Figure 4 — Components Object Model Overview

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.

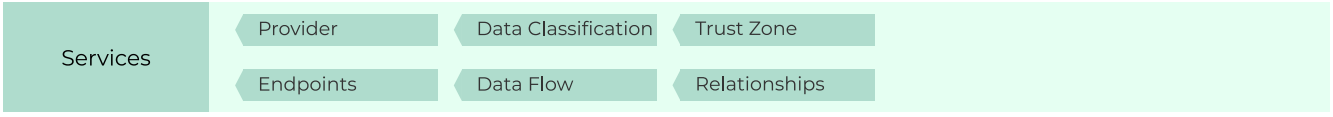


Figure 5 — Services Object Model Overview

© Ecma International 2025

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.



Figure 6 — Dependencies Object Model Overview

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.



Figure 7 — Compositions Object Model Overview

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.



Figure 8 — Vulnerabilities Object Model Overview

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.

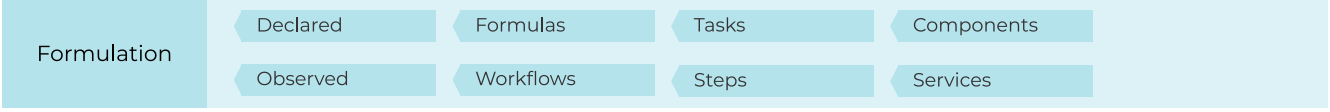


Figure 9 — Formulation Object Model Overview

© Ecma International 2025

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.

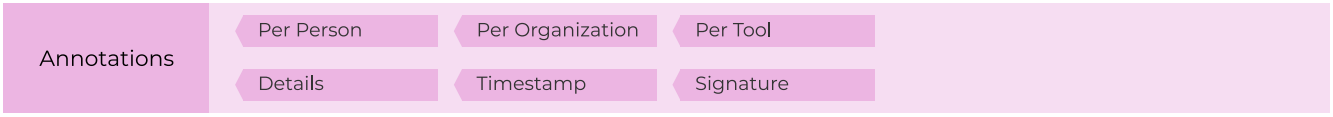


Figure 10 — Annotations Object Model Overview

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.



Figure 11 — Definitions Object Model Overview

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".



Figure 12 — Declarations Object Model Overview

Citations

Citations provide structured attributions within a BOM, identifying which entity or process supplied specific pieces of information. Each citation associates data with the component, service, tool, organization, person, or process through JSON Pointers or path expressions, over time. When combined with CycloneDX Attestations and/or Formulation, citations provide even greater assurance by linking attributions to evidence of integrity and to the processes that generated or transformed the data. This integration enables consumers to not only see who supplied the information, but also verify its authenticity and understand the context in which it was created.



Figure 13 — Citations Object Model Overview

© Ecma International 2025

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.



Figure 14 — Extensions Object Model Overview

Serialization Formats

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

Table 1 — Serialization formats in CycloneDX

Format	Resource	URL
JSON	Documentation	https://cyclonedx.org/docs/latest/json/
JSON	Schema	https://cyclonedx.org/schema/bom-1.7.schema.json
XML	Documentation	https://cyclonedx.org/docs/latest/xml/
XML	Schema	https://cyclonedx.org/schema/bom-1.7.xsd
Protobuf	Schema	https://cyclonedx.org/schema/bom-1.7.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

© Ecma International 2025

Table 2 — 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 shall 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 shall conform to RFC 4122. Use of serial numbers is recommended.
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 the formulation of any referencable object within the BOM, including components, services, metadata, declarations, or the BOM itself. This may encompass how the object was created, assembled, deployed, tested, certified, or otherwise brought into its present form. Common examples include software build pipelines, deployment processes, AI/ML model training, cryptographic key generation or certification, and third-party audits. Processes are modelled using declared and observed formulas, composed of workflows, tasks, and individual steps.

© Ecma International 2025

Table 2 — Properties for the root object *(continued)*

Property	Type	Requirement	Description
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.
citations	Array	Optional	A collection of attributions indicating which entity supplied information for specific fields within 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 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 (enum)

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 shall be "CycloneDX".

Enumeration of possible values:

- CycloneDX

6.2 CycloneDX Specification Version

Location: /specVersion
Property: specVersion (Required)
Type: String

The version of the CycloneDX specification the BOM conforms to.

Example (Informative)

```
1.7
```

6.3 BOM Serial Number

Location: /serialNumber
Property: serialNumber (Optional)
Type: String

© Ecma International 2025

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

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 shall conform to RFC 4122. Use of serial numbers is recommended.

Example (Informative)
`urn:uuid:3e671687-395b-41f5-a30f-a58921a69b79`

6.4 BOM Version

Location: /version
Property: version (Optional)
Type: Integer
Minimum Value: 1
Default Value: 1

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'.

Example (Informative)
`1`

6.5 BOM Metadata

Location: /metadata
Property: metadata (Optional)
Type: Object

Provides additional information about a BOM.

Table 3 — 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.

© Ecma International 2025

Table 3 — Properties for the metadata object *(continued)*

Property	Type	Requirement	Description
component	Array	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 licence information for the BOM document. This may be different from the licence(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.
distributionConstraints	Object	Optional	Conditions and constraints governing the sharing and distribution of the data or components described by this BOM.

6.5.1 Timestamp

Location: /metadata/timestamp
Property: timestamp (Optional)
Type: String
Format: data-time as specified in RFC 3339 section 5.6

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

6.5.2 Lifecycles

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

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 shall be a Lifecycle object.

6.5.2.1 Lifecycle

Location: /metadata/lifecycles/[]
Type: Object

The product lifecycle(s) that this BOM represents.

© Ecma International 2025

Shall be one of:

1. Pre-Defined Phase
2. Custom Phase

6.5.2.1.1 Pre-Defined Phase

Type: Object

Table 4 — Properties for the pre-defined phase object

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

6.5.2.1.2 Custom Phase

Type: Object

Table 5 — Properties for the custom phase object

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

6.5.2.1.3 Phase

Location: /metadata/lifecycles/[]/phase
Property: phase (Required)
Type: String (enum)

A pre-defined phase in the product lifecycle.

Table 6 — 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.

© Ecma International 2025

Table 6 — Enumeration of possible values *(continued)*

Value	Description
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.

6.5.2.1.4 Name

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

The name of the lifecycle phase.

6.5.2.1.5 Description

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

The description of the lifecycle phase.

6.5.3 Tools

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

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

Shall be one of:

1. Tools
2. Tools (legacy)

6.5.4 Tools

Type: Object

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

© Ecma International 2025

Table 7 — Properties for the tools object

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/[].

LEGACY

6.5.5 Tools (legacy)

Type: Array

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

6.5.6 Components

Location: /metadata/tools/components
Property: components (Optional)
Type: Array
Uniqueness: All items shall be unique.

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

6.5.6.1 Component

Location: /metadata/tools/components/[]
Type: Object

6.5.7 Services

Location: /metadata/tools/services
Property: services (Optional)
Type: Array
Uniqueness: All items shall be unique.

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 shall be a Service object.

© Ecma International 2025

6.5.7.1 Service

Location: /metadata/tools/services/[]
Type: Object

LEGACY

6.5.8 Tools (legacy)

Location: /metadata/tools
Property: tools
Type: Array

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

6.5.8.1 Tool

Location: /metadata/tools/[]
Type: Object

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

Table 8 — Properties for the tool 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.

6.5.8.1.1 Tool Vendor

Location: /metadata/tools/[]/vendor
Type: String

The name of the vendor who created the tool.

6.5.8.1.2 Tool Name

Location: /metadata/tools/[]/name
Type: String

The name of the tool.

© Ecma International 2025

6.5.8.1.3 Tool Version

Location: /metadata/tools/[]/version
Type: String

The version of the tool.

Example 1 (Informative)
```
9.0.14
```

Example 2 (Informative)
```
v1.33.7
```

Example 3 (Informative)
```
7.0.0-M1
```

Example 4 (Informative)
```
2.0pre1
```

Example 5 (Informative)
```
1.0.0-beta1
```

Example 6 (Informative)
```
0.8.15
```

6.5.8.1.4 Hashes

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

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

6.5.8.1.5 Hash

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

Table 9 — Properties for the hash object

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

© Ecma International 2025

6.5.8.1.6 Hash Algorithm

Location: /metadata/tools/[]/hashes/[]/alg
Type: String (enum)

The algorithm that generated the hash value.

Enumeration of possible values:

- MD5
- SHA-1
- SHA-256
- SHA-384
- SHA-512
- SHA3-256
- SHA3-384
- SHA3-512
- BLAKE2b-256
- BLAKE2b-384
- BLAKE2b-512
- BLAKE3
- Streebog-256
- Streebog-512

6.5.8.1.7 Hash Value

Location: /metadata/tools/[]/hashes/[]/content
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})`

The value of the hash.

Example (Informative)
`3942447fac867ae5cdb3229b658f4d48`

6.5.8.1.8 External References

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

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 shall be an External Reference object.

6.5.8.1.9 External Reference

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

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.

© Ecma International 2025

6.5.9 BOM Manufacturer

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

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 10 — Properties for the manufacturer object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.5.9.1 BOM Reference

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

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

6.5.9.2 Organization Name

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

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.5.9.3 Organization Address

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

The physical address (location) of the organization.

© Ecma International 2025

Table 11 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.5.9.3.1 BOM Reference

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

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

6.5.9.3.2 Country

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

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

6.5.9.3.3 Region

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

The region or state in the country.

Example (Informative)

```
Texas
```


© Ecma International 2025

6.5.9.3.4 Locality

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

The locality or city within the country.

Example (Informative)
```
Austin
```

6.5.9.3.5 Post Office Box Number

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

The post office box number.

Example (Informative)
```
901
```

6.5.9.3.6 Postal Code

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

The postal code.

Example (Informative)
```
78758
```

6.5.9.3.7 Street Address

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

The street address.

Example (Informative)
```
100 Main Street
```

© Ecma International 2025

6.5.9.4 Organization URL(s)

Location: /metadata/manufacturer/url
Property: url (Optional)
Type: Array (of String)
Format: iri-reference as specified in RFC 3987

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

Example (Informative)
`https://example.com`

6.5.9.5 Organizational Contact

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

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.5.9.5.1 Organizational Person

Location: /metadata/manufacturer/contact/[]
Type: Object

Table 12 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.5.9.5.2 BOM Reference

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

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

© Ecma International 2025

6.5.9.5.3 Name

Location: /metadata/manufacturer/contact/[]/name
Type: String

The name of a contact.

Example (Informative)
```
Contact name
```

6.5.9.5.4 Email Address

Location: /metadata/manufacturer/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.5.9.5.5 Phone

Location: /metadata/manufacturer/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.5.10 BOM Authors

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

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 shall be an Organizational Person object.

6.5.10.1 Organizational Person

Location: /metadata/authors/[]
Type: Object

© Ecma International 2025

Table 13 — Properties for the author object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.5.10.1.1 BOM Reference

Location: /metadata/authors/[]/bom-ref
Type: String

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

6.5.10.1.2 Name

Location: /metadata/authors/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.5.10.1.3 Email Address

Location: /metadata/authors/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

6.5.10.1.4 Phone

Location: /metadata/authors/[]/phone
Type: String

The phone number of the contact.

Example (Informative)

```
800-555-1212
```


© Ecma International 2025

6.5.11 Component

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

The component that the BOM describes.

LEGACY

6.5.12 Component Manufacture (legacy)

Location: /metadata/manufacture
Property: manufacture (Optional)
Type: Object

[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 14 — Properties for the manufacture object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.5.12.1 BOM Reference

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

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

6.5.12.2 Organization Name

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

The name of the organization.

Example (Informative)

```
Example Inc.
```

© Ecma International 2025

6.5.12.3 Organization Address

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

The physical address (location) of the organization.

Table 15 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.5.12.3.1 BOM Reference

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

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

6.5.12.3.2 Country

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

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

© Ecma International 2025

6.5.12.3.3 Region

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

The region or state in the country.

Example (Informative)
```
Texas
```

6.5.12.3.4 Locality

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

The locality or city within the country.

Example (Informative)
```
Austin
```

6.5.12.3.5 Post Office Box Number

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

The post office box number.

Example (Informative)
```
901
```

6.5.12.3.6 Postal Code

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

The postal code.

Example (Informative)
```
78758
```

© Ecma International 2025

6.5.12.3.7 Street Address

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

The street address.

Example (Informative)
```
100 Main Street
```

6.5.12.4 Organization URL(s)

Location: /metadata/manufacture/url
Property: url (Optional)
Type: Array (of String)
Format: iri-reference as specified in RFC 3987

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

Example (Informative)
```
https://example.com
```

6.5.12.5 Organizational Contact

Location: /metadata/manufacture/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.5.12.5.1 Organizational Person

Location: /metadata/manufacture/contact/[]
Type: Object

Table 16 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

© Ecma International 2025

6.5.12.5.2 BOM Reference

Location: /metadata/manufacture/contact/[]/bom-ref
Type: String

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

6.5.12.5.3 Name

Location: /metadata/manufacture/contact/[]/name
Type: String

The name of a contact.

Example (Informative)
```
Contact name
```

6.5.12.5.4 Email Address

Location: /metadata/manufacture/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.5.12.5.5 Phone

Location: /metadata/manufacture/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.5.13 Supplier

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

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.

© Ecma International 2025

Table 17 — Properties for the supplier object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.5.13.1 BOM Reference

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

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

6.5.13.2 Organization Name

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

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.5.13.3 Organization Address

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

The physical address (location) of the organization.

Table 18 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

© Ecma International 2025

Table 18 — Properties for the address object *(continued)*

Property	Type	Requirement	Description
postOfficeBoxNumber	String	Optional	The post office box number.
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

6.5.13.3.1 BOM Reference

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

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

6.5.13.3.2 Country

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

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

6.5.13.3.3 Region

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

The region or state in the country.

Example (Informative)

```
Texas
```

6.5.13.3.4 Locality

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

The locality or city within the country.

Example (Informative)

```
Austin
```

© Ecma International 2025

6.5.13.3.5 Post Office Box Number

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

The post office box number.

Example (Informative)
```
901
```

6.5.13.3.6 Postal Code

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

The postal code.

Example (Informative)
```
78758
```

6.5.13.3.7 Street Address

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

The street address.

Example (Informative)
```
100 Main Street
```

6.5.13.4 Organization URL(s)

Location: /metadata/supplier/url
Property: url (Optional)
Type: Array (of String)
Format: iri-reference as specified in RFC 3987

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

Example (Informative)
```
https://example.com
```

6.5.13.5 Organizational Contact

Location: /metadata/supplier/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

© Ecma International 2025

6.5.13.5.1 Organizational Person

Location: /metadata/supplier/contact/[]
Type: Object

Table 19 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.5.13.5.2 BOM Reference

Location: /metadata/supplier/contact/[]/bom-ref
Type: String

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

6.5.13.5.3 Name

Location: /metadata/supplier/contact/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.5.13.5.4 Email Address

Location: /metadata/supplier/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

© Ecma International 2025

6.5.13.5.5 Phone

Location: /metadata/supplier/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)
800-555-1212

6.5.14 BOM License(s)

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

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

6.5.14.1 License

Location: /metadata/licenses/[]

6.5.15 Properties

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

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 shall be a Lightweight name-value pair object.

6.5.15.1 Lightweight name-value pair

Location: /metadata/properties/[]
Type: Object

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 20 — Properties for the property 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.

© Ecma International 2025

6.5.15.1.1 Name

Location: /metadata/properties/[]/name
Type: String

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

6.5.15.1.2 Value

Location: /metadata/properties/[]/value
Type: String

The value of the property.

6.5.16 Distribution Constraints

Location: /metadata/distributionConstraints
Property: distributionConstraints (Optional)
Type: Object

Conditions and constraints governing the sharing and distribution of the data or components described by this BOM.

Table 21 — Properties for the distributionConstraints object

Property	Type	Requirement	Description
tlp	String	Optional	The Traffic Light Protocol (TLP) classification that controls the sharing and distribution of the data that the BOM describes.

6.5.16.1 Traffic Light Protocol (TLP) Classification

Location: /metadata/distributionConstraints/tlp
Property: tlp (Optional)
Type: String (enum)
Default Value: CLEAR

The Traffic Light Protocol (TLP) classification that controls the sharing and distribution of the data that the BOM describes.

Table 22 — Enumeration of possible values

Value	Description
CLEAR	The information is not subject to any restrictions as regards the sharing.
GREEN	The information is subject to limited disclosure, and recipients can share it within their community but not via publicly accessible channels.
AMBER	The information is subject to limited disclosure, and recipients can only share it on a need-to-know basis within their organization and with clients.
AMBER_AND_STRICT	The information is subject to limited disclosure, and recipients can only share it on a need-to-know basis within their organization.
RED	The information is subject to restricted distribution to individual recipients only and shall not be shared.

© Ecma International 2025

6.6 Components

Location: /components
Property: components (Optional)
Type: Array
Uniqueness: All items shall be unique.

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

6.6.1 Component

Location: /components/[]
Type: Object

Table 23 — Properties for the component 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 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 identifier which can be used to reference the component elsewhere in the BOM. Every **`bom-ref`** shall 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.
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.

© Ecma International 2025

Table 23 — Properties for the component object *(continued)*

Property	Type	Requirement	Description
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. Shall be used exclusively, either 'version' or 'versionRange', but not both.
versionRange	String	Optional	For an external component, this specifies the accepted version range. The value shall adhere to the Package URL Version Range syntax (vers), as defined at https://github.com/package-url/vers-spec May only be used if **`.isExternal`** is set to **`true`**. Shall be used exclusively, either 'version' or 'versionRange', but not both.
isExternal	Boolean	Optional	Determine whether this component is external. An external component is one that is not part of an assembly, but is expected to be provided by the environment, regardless of the component's **`.scope`**. This setting can be useful for distinguishing which components are bundled with the product and which can be relied upon to be present in the deployment environment. This may be set to **`true`** for runtime components only. For **`$.metadata.component`**, it shall be set to **`false`**.
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	A list of SPDX licenses and/or named licenses and/or SPDX Licence Expression.
copyright	String	Optional	A copyright notice informing users of the underlying claims to copyright ownership in a published work.
patentAssertions	Array	Optional	A list of assertions made regarding patents associated with this component or service. Assertions distinguish between ownership, licensing, and other relevant interactions with patents.
cpe	String	Optional	Asserts the identity of the component using CPE. The CPE shall 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, shall 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.

© Ecma International 2025

Table 23 — Properties for the component object *(continued)*

Property	Type	Requirement	Description
omniborId	Array	Optional	Asserts the identity of the component using the OmniBOR Artifact ID. The OmniBOR, if specified, shall 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.
swhid	Array	Optional	Asserts the identity of the component using the Software Heritage persistent identifier (SWHID). The SWHID, if specified, shall 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 release notes.

© Ecma International 2025

Table 23 — Properties for the component object *(continued)*

Property	Type	Requirement	Description
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 shall not be specified for other component types.
data	Array	Optional	This object should be specified for any component of type **`data`** and shall 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).

6.6.1.1 Component Type

Location: /components/[]/type
Type: String (enum)

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.

Example (Informative)

```
library
```

© Ecma International 2025

Table 24 — 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. 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 that interprets or executes software. This may include runtimes such as those that execute bytecode, just-in-time compilers, interpreters, 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.

© Ecma International 2025

6.6.1.2 Mime-Type

Location: /components/[]/mime-type
Type: String
Pattern Constraint: `^[-+a-z0-9.]+/[-+a-z0-9.]+$`

The 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.

Example (Informative)
`image/jpeg`

6.6.1.3 BOM Reference

Location: /components/[]/bom-ref
Type: String

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

6.6.1.4 Component Supplier

Location: /components/[]/supplier
Type: Object

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

Table 25 — Properties for the supplier object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.4.1 BOM Reference

Location: /components/[]/supplier/bom-ref
Type: String

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

© Ecma International 2025

6.6.1.4.2 Organization Name

Location: /components/[]/supplier/name
Type: String

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.6.1.4.3 Organization Address

Location: /components/[]/supplier/address
Type: Object

The physical address (location) of the organization.

Table 26 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.4.4 BOM Reference

Location: /components/[]/supplier/address/bom-ref
Type: String

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

6.6.1.4.5 Country

Location: /components/[]/supplier/address/country
Type: String

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

© Ecma International 2025

6.6.1.4.6 Region

Location: /components/[]/supplier/address/region
Type: String

The region or state in the country.

Example (Informative)
```
Texas
```

6.6.1.4.7 Locality

Location: /components/[]/supplier/address/locality
Type: String

The locality or city within the country.

Example (Informative)
```
Austin
```

6.6.1.4.8 Post Office Box Number

Location: /components/[]/supplier/address/postOfficeBoxNumber
Type: String

The post office box number.

Example (Informative)
```
901
```

6.6.1.4.9 Postal Code

Location: /components/[]/supplier/address/postalCode
Type: String

The postal code.

Example (Informative)
```
78758
```

6.6.1.4.10 Street Address

Location: /components/[]/supplier/address/streetAddress
Type: String

The street address.

Example (Informative)
```
100 Main Street
```

© Ecma International 2025

6.6.1.4.11 Organization URL(s)

Location: /components/[]/supplier/url
Property: url (Optional)
Type: Array (of String)
Format: iri-reference as specified in RFC 3987

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

Example (Informative)

```
https://example.com
```

6.6.1.4.12 Organizational Contact

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

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.6.1.4.13 Organizational Person

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

Table 27 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.4.14 BOM Reference

Location: /components/[]/supplier/contact/[]/bom-ref
Type: String

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

6.6.1.4.15 Name

Location: /components/[]/supplier/contact/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```


© Ecma International 2025

6.6.1.4.16 Email Address

Location: /components/[]/supplier/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
`firstname.lastname@example.com`

6.6.1.4.17 Phone

Location: /components/[]/supplier/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)
`800-555-1212`

6.6.1.5 Component Manufacturer

Location: /components/[]/manufacturer
Type: Object

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 28 — Properties for the manufacturer object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.5.1 BOM Reference

Location: /components/[]/manufacturer/bom-ref
Type: String

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

© Ecma International 2025

6.6.1.5.2 Organization Name

Location: /components/[]/manufacturer/name
Type: String

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.6.1.5.3 Organization Address

Location: /components/[]/manufacturer/address
Type: Object

The physical address (location) of the organization.

Table 29 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.5.4 BOM Reference

Location: /components/[]/manufacturer/address/bom-ref
Type: String

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

6.6.1.5.5 Country

Location: /components/[]/manufacturer/address/country
Type: String

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

© Ecma International 2025

6.6.1.5.6 Region

Location: /components/[]/manufacturer/address/region
Type: String

The region or state in the country.

Example (Informative)
```
Texas
```

6.6.1.5.7 Locality

Location: /components/[]/manufacturer/address/locality
Type: String

The locality or city within the country.

Example (Informative)
```
Austin
```

6.6.1.5.8 Post Office Box Number

Location: /components/[]/manufacturer/address/postOfficeBoxNumber
Type: String

The post office box number.

Example (Informative)
```
901
```

6.6.1.5.9 Postal Code

Location: /components/[]/manufacturer/address/postalCode
Type: String

The postal code.

Example (Informative)
```
78758
```

6.6.1.5.10 Street Address

Location: /components/[]/manufacturer/address/streetAddress
Type: String

The street address.

Example (Informative)
```
100 Main Street
```

© Ecma International 2025

6.6.1.5.11 Organization URL(s)

Location: /components/[]/manufacturer/url
Property: url (Optional)
Type: Array (of String)
Format: iri-reference as specified in RFC 3987

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

Example (Informative)
```
https://example.com
```

6.6.1.5.12 Organizational Contact

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

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.6.1.5.13 Organizational Person

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

Table 30 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.5.14 BOM Reference

Location: /components/[]/manufacturer/contact/[]/bom-ref
Type: String

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

6.6.1.5.15 Name

Location: /components/[]/manufacturer/contact/[]/name
Type: String

The name of a contact.

Example (Informative)
```
Contact name
```


© Ecma International 2025

6.6.1.5.16 Email Address

Location: /components/[]/manufacturer/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.6.1.5.17 Phone

Location: /components/[]/manufacturer/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.6.1.6 Component Authors

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

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 shall be an Organizational Person object.

6.6.1.6.1 Organizational Person

Location: /components/[]/authors/[]
Type: Object

Table 31 — Properties for the author object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

© Ecma International 2025

6.6.1.6.2 BOM Reference

Location: /components/[]/authors/[]/bom-ref
Type: String

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

6.6.1.6.3 Name

Location: /components/[]/authors/[]/name
Type: String

The name of a contact.

Example (Informative)
```
Contact name
```

6.6.1.6.4 Email Address

Location: /components/[]/authors/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.6.1.6.5 Phone

Location: /components/[]/authors/[]/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

LEGACY

6.6.1.7 Component Author (legacy)

Location: /components/[]/author
Type: String

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

Example (Informative)
```
Acme Inc
```


© Ecma International 2025

6.6.1.8 Component Publisher

Location: /components/[]/publisher
Type: String

The person(s) or organization(s) that published the component.

Example (Informative)
```
Acme Inc
```

6.6.1.9 Component Group

Location: /components/[]/group
Type: String

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.

Example (Informative)
```
com.acme
```

6.6.1.10 Component Name

Location: /components/[]/name
Type: String

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

Example (Informative)
```
tomcat-catalina
```

6.6.1.11 Component Version

Location: /components/[]/version
Type: String

The component version. The version should ideally comply with semantic versioning but is not enforced. Shall be used exclusively, either 'version' or 'versionRange', but not both.

Example 1 (Informative)
```
9.0.14
```

Example 2 (Informative)
```
v1.33.7
```

Example 3 (Informative)
```
7.0.0-M1
```

Example 4 (Informative)
```
2.0pre1
```

© Ecma International 2025

Example 5 (Informative)
```
1.0.0-beta1
```

Example 6 (Informative)
```
0.8.15
```

6.6.1.12 Component Version Range

Location: /components/[]/versionRange
Type: String

For an external component, this specifies the accepted version range. The value shall adhere to the Package URL Version Range syntax (vers), as defined at https://github.com/package-url/vers-spec May only be used if **`.isExternal`** is set to **`true`**. Shall be used exclusively, either 'version' or 'versionRange', but not both.

Example 1 (Informative)
```
vers:cargo/9.0.14
```

Example 2 (Informative)
```
vers:npm/1.2.3|>=2.0.0|<5.0.0
```

Example 3 (Informative)
```
vers:pypi/0.0.0|0.0.1|0.0.2|0.0.3|1.0|2.0pre1
```

Example 4 (Informative)
```
ers: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.
```

Example 5 (Informative)
```
vers:gem/>=2.2.0|!= 2.2.1|<2.3.0
```

6.6.1.13 Component Is External

Location: /components/[]/isExternal
Type: Boolean

Determine whether this component is external. An external component is one that is not part of an assembly, but is expected to be provided by the environment, regardless of the component's **`.scope`**. This setting can be useful for distinguishing which components are bundled with the product and which can be relied upon to be present in the deployment environment. This may be set to **`true`** for runtime components only. For **`$.metadata.component`**, it shall be set to **`false`**.

6.6.1.14 Component Description

Location: /components/[]/description
Type: String

Specifies a description for the component.

© Ecma International 2025

6.6.1.15 Component Scope

Location: /components/[]/scope
Type: String (enum)
Default Value: required

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

Table 32 — 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 shall 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.

6.6.1.16 Component Hashes

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

The hashes of the component. Each item of this array shall be a Hash object.

6.6.1.16.1 Hash

Location: /components/[]/hashes/[]
Type: Object

Table 33 — Properties for the hash object

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

6.6.1.16.2 Hash Algorithm

Location: /components/[]/hashes/[]/alg
Type: String (enum)

The algorithm that generated the hash value.

Enumeration of possible values:

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

© Ecma International 2025

- SHA3-256
- SHA3-384
- SHA3-512
- BLAKE2b-256
- BLAKE2b-384
- BLAKE2b-512
- BLAKE3
- Streebog-256
- Streebog-512

6.6.1.16.3 Hash Value

Location: /components/[]/hashes/[]/content
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})$`

The value of the hash.

Example (Informative)
`3942447fac867ae5cdb3229b658f4d48`

6.6.1.17 Component License(s)

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

A list of SPDX licenses and/or named licenses and/or SPDX Licence Expression.

6.6.1.17.1 License

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

Shall be one of:

1. Licence
2. Licence Expression

6.6.1.17.2 License

Type: Object

Table 34 — Properties for the license object

Property	Type	Requirement	Description
license	Object	Required	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 licensing information, and the full text of the licence.

© Ecma International 2025

6.6.1.17.3 License Expression

Type: Object

Specifies the details and attributes related to a software licence. It shall be a valid SPDX licence expression, along with additional properties such as licence acknowledgment.

Table 35 — Properties for the license expression object

Property	Type	Requirement	Description
expression	String	Required	A valid SPDX licence expression. Refer to https://spdx.org/specifications for syntax requirements.
expressionDetails	Array	Optional	Details for parts of the **`expression`**.
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 licence.
bom-ref	String	Optional	An identifier which can be used to reference the licence elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. Value should not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
licensing	Object	Optional	Licensing details describing the licensor/licensee, licence 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.

6.6.1.17.4 License

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

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 licensing information, and the full text of the licence.

© Ecma International 2025

Table 36 — Properties for the license object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the licence elsewhere in the BOM. Every **`bom-ref`** shall 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 licence identifier. If specified, this value shall 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.
name	String	Optional	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.
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 licence.
text	Object	Optional	A way to include the textual content of a licence.
url	String	Optional	The URL to the licence file. If specified, a 'licence' externalReference should also be specified for completeness.
licensing	Object	Optional	Licensing details describing the licensor/licensee, licence 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.

6.6.1.17.5 BOM Reference

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

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

© Ecma International 2025

6.6.1.17.6 License ID (SPDX)

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

A valid SPDX licence identifier. If specified, this value shall 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.

Example (Informative)

```
Apache-2.0
```

6.6.1.17.7 License Name

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

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.

Example (Informative)

```
Acme Software License
```

6.6.1.17.8 License Acknowledgement

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

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 licence.

Table 37 — Enumeration of possible values

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

6.6.1.17.9 License text

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

A way to include the textual content of a licence.

© Ecma International 2025

Table 38 — 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 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.

6.6.1.17.10 Content-Type

Location: /components/[]/licenses/[]/license/text/contentType
Property: contentType (Optional)
Type: String
Default Value: text/plain

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.

Example 1 (Informative)

```
text/plain
```

Example 2 (Informative)

```
application/json
```

Example 3 (Informative)

```
image/png
```

6.6.1.17.11 Encoding

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

Specifies the encoding the text is represented in.

Table 39 — Enumeration of possible values

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

© Ecma International 2025

6.6.1.17.12 Attachment Text

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

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

6.6.1.17.13 License URL

Location: /components/[]/licenses/[]/license/url
Property: url (Optional)
Type: String
Format: iri-reference as specified in RFC 3987

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

Example (Informative)

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

6.6.1.17.14 Licensing information

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

Licensing details describing the licensor/licensee, licence type, renewal and expiration dates, and other important metadata.

Table 40 — Properties for the licensing object

Property	Type	Requirement	Description
altIds	Array	Optional	Licence identifiers that may be used to manage licenses and their lifecycle.
licensor	Array	Optional	The individual or organization that grants a licence to another individual or organization.
licensee	Array	Optional	The individual or organization for which a licence was granted to.
purchaser	Array	Optional	The individual or organization that purchased the licence.
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 licence(s) that was granted to the licensee.
lastRenewal	String	Optional	The timestamp indicating when the licence 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 licence was last renewed.
expiration	String	Optional	The timestamp indicating when the current licence expires (if applicable).

© Ecma International 2025

6.6.1.17.15 Alternate License Identifiers

Location: /components/[]/licenses/[]/license/licensing/altIds
Property: altIds (Optional)
Type: Array (of String)

Licence identifiers that may be used to manage licenses and their lifecycle Each item of this array shall be a string.

6.6.1.17.16 Licensor

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

The individual or organization that grants a licence to another individual or organization.

Table 41 — Properties for the licensor object

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

6.6.1.17.17 Licensor (Organization)

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

The organization that granted the licence.

Table 42 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.17.18 BOM Reference

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

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

© Ecma International 2025

6.6.1.17.19 Organization Name

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

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.6.1.17.20 Organization Address

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

The physical address (location) of the organization.

Table 43 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.17.21 BOM Reference

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

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

6.6.1.17.22 Country

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

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

© Ecma International 2025

6.6.1.17.23 Region

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

The region or state in the country.

Example (Informative)
```
Texas
```

6.6.1.17.24 Locality

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

The locality or city within the country.

Example (Informative)
```
Austin
```

6.6.1.17.25 Post Office Box Number

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

The post office box number.

Example (Informative)
```
901
```

6.6.1.17.26 Postal Code

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

The postal code.

Example (Informative)
```
78758
```

6.6.1.17.27 Street Address

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

The street address.

Example (Informative)
```
100 Main Street
```


© Ecma International 2025

6.6.1.17.28 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

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

Example (Informative)

```
https://example.com
```

6.6.1.17.29 Organizational Contact

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

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.6.1.17.30 Organizational Person

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

Table 44 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.17.31 BOM Reference

Location: /components/[]/licenses/[]/license/licensing/licensor/organization/contact/[]/bom-ref
Type: String

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

6.6.1.17.32 Name

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

The name of a contact.

Example (Informative)

```
Contact name
```

© Ecma International 2025

6.6.1.17.33 Email Address

Location: /components/[]/licenses/[]/license/licensing/licensor/organization/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
`firstname.lastname@example.com`

6.6.1.17.34 Phone

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

The phone number of the contact.

Example (Informative)
`800-555-1212`

6.6.1.17.35 Licensor (Individual)

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

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

Table 45 — Properties for the individual object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.17.36 BOM Reference

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

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

© Ecma International 2025

6.6.1.17.37 Name

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

The name of a contact.

Example (Informative)
```
Contact name
```

6.6.1.17.38 Email Address

Location: /components/[]/licenses/[]/license/licensing/licensor/individual/email
Property: email (Optional)
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.6.1.17.39 Phone

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

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.6.1.17.40 Licensee

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

The individual or organization for which a licence was granted to.

Table 46 — Properties for the licensee object

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

© Ecma International 2025

6.6.1.17.41 Licensee (Organization)

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

The organization that was granted the licence.

Table 47 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.17.42 BOM Reference

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

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

6.6.1.17.43 Organization Name

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

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.6.1.17.44 Organization Address

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

The physical address (location) of the organization.

© Ecma International 2025

Table 48 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.17.45 BOM Reference

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

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

6.6.1.17.46 Country

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

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

6.6.1.17.47 Region

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

The region or state in the country.

Example (Informative)

```
Texas
```

© Ecma International 2025

6.6.1.17.48 Locality

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

The locality or city within the country.

Example (Informative)
```
Austin
```

6.6.1.17.49 Post Office Box Number

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

The post office box number.

Example (Informative)
```
901
```

6.6.1.17.50 Postal Code

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

The postal code.

Example (Informative)
```
78758
```

6.6.1.17.51 Street Address

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

The street address.

Example (Informative)
```
100 Main Street
```

6.6.1.17.52 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

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

Example (Informative)
```
https://example.com
```


© Ecma International 2025

6.6.1.17.53 Organizational Contact

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

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.6.1.17.54 Organizational Person

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

Table 49 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.17.55 BOM Reference

Location: /components/[]/licenses/[]/license/licensing/licensee/organization/contact/[]/bom-ref
Type: String

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

6.6.1.17.56 Name

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

The name of a contact.

Example (Informative)

```
Contact name
```

6.6.1.17.57 Email Address

Location: /components/[]/licenses/[]/license/licensing/licensee/organization/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

© Ecma International 2025

6.6.1.17.58 Phone

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

The phone number of the contact.

Example (Informative)

```
800-555-1212
```

6.6.1.17.59 Licensee (Individual)

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

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

Table 50 — Properties for the individual object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.17.60 BOM Reference

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

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

6.6.1.17.61 Name

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

The name of a contact.

Example (Informative)

```
Contact name
```


© Ecma International 2025

6.6.1.17.62 Email Address

Location: /components/[]/licenses/[]/license/licensing/licensee/individual/email
Property: email (Optional)
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

6.6.1.17.63 Phone

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

The phone number of the contact.

Example (Informative)

```
800-555-1212
```

6.6.1.17.64 Purchaser

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

The individual or organization that purchased the licence.

Table 51 — Properties for the purchaser object

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

6.6.1.17.65 Purchaser (Organization)

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

The organization that purchased the licence.

© Ecma International 2025

Table 52 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.17.66 BOM Reference

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

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

6.6.1.17.67 Organization Name

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

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.6.1.17.68 Organization Address

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

The physical address (location) of the organization.

Table 53 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

© Ecma International 2025

Table 53 — Properties for the address object *(continued)*

Property	Type	Requirement	Description
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

6.6.1.17.69 BOM Reference

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

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

6.6.1.17.70 Country

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

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

6.6.1.17.71 Region

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

The region or state in the country.

Example (Informative)

```
Texas
```

6.6.1.17.72 Locality

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

The locality or city within the country.

Example (Informative)

```
Austin
```

© Ecma International 2025

6.6.1.17.73 Post Office Box Number

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

The post office box number.

Example (Informative)
```
901
```

6.6.1.17.74 Postal Code

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

The postal code.

Example (Informative)
```
78758
```

6.6.1.17.75 Street Address

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

The street address.

Example (Informative)
```
100 Main Street
```

6.6.1.17.76 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

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

Example (Informative)
```
https://example.com
```

6.6.1.17.77 Organizational Contact

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

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

© Ecma International 2025

6.6.1.17.78 Organizational Person

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

Table 54 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.17.79 BOM Reference

Location: /components/[]/licenses/[]/license/licensing/purchaser/organization/contact/[]/bom-ref
Type: String

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

6.6.1.17.80 Name

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

The name of a contact.

Example (Informative)

```
Contact name
```

6.6.1.17.81 Email Address

Location: /components/[]/licenses/[]/license/licensing/purchaser/organization/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

© Ecma International 2025

6.6.1.17.82 Phone

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

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.6.1.17.83 Purchaser (Individual)

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

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

Table 55 — Properties for the individual object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.17.84 BOM Reference

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

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

6.6.1.17.85 Name

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

The name of a contact.

Example (Informative)
```
Contact name
```


© Ecma International 2025

6.6.1.17.86 Email Address

Location: /components/[]/licenses/[]/license/licensing/purchaser/individual/email
Property: email (Optional)
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.6.1.17.87 Phone

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

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.6.1.17.88 Purchase Order

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

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

6.6.1.17.89 License Type

Location: /components/[]/licenses/[]/license/licensing/licenseTypes
Property: licenseTypes (Optional)
Type: Array (of String) (enum)

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

Table 56 — 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 licenses for a software application and licenses 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.

© Ecma International 2025

Table 56 — Enumeration of possible values *(continued)*

Value	Description
evaluation	A licence that grants permission to install 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.
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.

6.6.1.17.90 Last Renewal

Location: /components/[]/licenses/[]/license/licensing/lastRenewal
Property: lastRenewal (Optional)
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The timestamp indicating when the licence 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 licence was last renewed.

6.6.1.17.91 Expiration

Location: /components/[]/licenses/[]/license/licensing/expiration
Property: expiration (Optional)
Type: String
Format: data-time as specified in RFC 3339 section 5.6

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

6.6.1.17.92 Properties

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

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 shall be a Lightweight name-value pair object.

© Ecma International 2025

6.6.1.17.93 Lightweight name-value pair

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

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 57 — Properties for the property 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.

6.6.1.17.94 Name

Location: /components/[]/licenses/[]/license/properties/[]/name
Type: String

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

6.6.1.17.95 Value

Location: /components/[]/licenses/[]/license/properties/[]/value
Type: String

The value of the property.

6.6.1.17.96 SPDX License Expression

Location: /components/[]/licenses/[]/expression
Property: expression (Required)
Type: String

A valid SPDX licence expression. Refer to https://spdx.org/specifications for syntax requirements.

Example 1 (Informative)

```
Apache-2.0 AND (MIT OR GPL-2.0-only)
```

Example 2 (Informative)

```
GPL-3.0-only WITH Classpath-exception-2.0
```

6.6.1.17.97 Expression Details

Location: /components/[]/licenses/[]/expressionDetails
Property: expressionDetails (Optional)
Type: Array

Details for parts of the **`expression`**.

© Ecma International 2025

6.6.1.17.98 ExpressionDetail

Location: /components/[]/licenses/[]/expressionDetails/[]
Type: Object

This document specifies the details and attributes related to a software licence identifier. An SPDX expression may be a compound of licence identifiers. The **`license_identifier`** property serves as the key that identifies each record. Note that this key is not required to be unique, as the same licence identifier could apply to multiple, different but similar licence details, texts, etc.

Table 58 — Properties for the expressionDetail object

Property	Type	Requirement	Description
licenseIdentifier	String	Required	The valid SPDX licence identifier. Refer to https://spdx.org/ specifications for syntax requirements. This property serves as the primary key, which uniquely identifies each record.
bom-ref	String	Optional	An identifier which can be used to reference the licence elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. Value should not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.
text	Object	Optional	A way to include the textual content of the licence.
url	String	Optional	The URL to the licence file. If specified, a 'licence' externalReference should also be specified for completeness.

6.6.1.17.99 License Identifier

Location: /components/[]/licenses/[]/expressionDetails/[]/licenseIdentifier
Type: String

The valid SPDX licence identifier. Refer to https://spdx.org/specifications for syntax requirements. This property serves as the primary key, which uniquely identifies each record.

Example 1 (Informative)

```
Apache-2.0
```

Example 2 (Informative)

```
GPL-3.0-only WITH Classpath-exception-2.0
```

Example 3 (Informative)

```
LicenseRef-my-custom-license
```

6.6.1.17.100 BOM Reference

Location: /components/[]/licenses/[]/expressionDetails/[]/bom-ref
Type: String

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

© Ecma International 2025

6.6.1.17.101 License texts

Location: /components/[]/licenses/[]/expressionDetails/[]/text
Type: Object

A way to include the textual content of the licence.

Table 59 — 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 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.

6.6.1.17.102 Content-Type

Location: /components/[]/licenses/[]/expressionDetails/[]/text/contentType
Type: String
Default Value: text/plain

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.

Example 1 (Informative)
```
text/plain
```

Example 2 (Informative)
```
application/json
```

Example 3 (Informative)
```
image/png
```

6.6.1.17.103 Encoding

Location: /components/[]/licenses/[]/expressionDetails/[]/text/encoding
Type: String (enum)

Specifies the encoding the text is represented in.

Table 60 — Enumeration of possible values

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

© Ecma International 2025

6.6.1.17.104 Attachment Text

Location: /components/[]/licenses/[]/expressionDetails/[]/text/content
Type: String

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

6.6.1.17.105 License URL

Location: /components/[]/licenses/[]/expressionDetails/[]/url
Type: String
Format: iri-reference as specified in RFC 3987

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

Example (Informative)

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

6.6.1.17.106 License Acknowledgement

Location: /components/[]/licenses/[]/acknowledgement
Property: acknowledgement (Optional)
Type: String (enum)

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 licence.

Table 61 — Enumeration of possible values

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

6.6.1.17.107 BOM Reference

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

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

© Ecma International 2025

6.6.1.17.108 Licensing information

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

Licensing details describing the licensor/licensee, licence type, renewal and expiration dates, and other important metadata.

Table 62 — Properties for the licensing object

Property	Type	Requirement	Description
altIds	Array	Optional	Licence identifiers that may be used to manage licenses and their lifecycle.
licensor	Array	Optional	The individual or organization that grants a licence to another individual or organization.
licensee	Array	Optional	The individual or organization for which a licence was granted to.
purchaser	Array	Optional	The individual or organization that purchased the licence.
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 licence(s) that was granted to the licensee.
lastRenewal	String	Optional	The timestamp indicating when the licence 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 licence was last renewed.
expiration	String	Optional	The timestamp indicating when the current licence expires (if applicable).

6.6.1.17.109 Alternate License Identifiers

Location: /components/[]/licenses/[]/licensing/altIds
Property: altIds (Optional)
Type: Array (of String)

Licence identifiers that may be used to manage licenses and their lifecycle Each item of this array shall be a string.

6.6.1.17.110 Licensor

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

The individual or organization that grants a licence to another individual or organization.

Table 63 — Properties for the licensor object

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

© Ecma International 2025

6.6.1.17.111 Licensor (Organization)

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

The organization that granted the licence.

Table 64 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.17.112 BOM Reference

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

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

6.6.1.17.113 Organization Name

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

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.6.1.17.114 Organization Address

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

The physical address (location) of the organization.

© Ecma International 2025

Table 65 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.17.115 BOM Reference

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

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

6.6.1.17.116 Country

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

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

6.6.1.17.117 Region

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

The region or state in the country.

Example (Informative)

```
Texas
```

© Ecma International 2025

6.6.1.17.118 Locality

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

The locality or city within the country.

Example (Informative)
```
Austin
```

6.6.1.17.119 Post Office Box Number

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

The post office box number.

Example (Informative)
```
901
```

6.6.1.17.120 Postal Code

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

The postal code.

Example (Informative)
```
78758
```

6.6.1.17.121 Street Address

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

The street address.

Example (Informative)
```
100 Main Street
```

6.6.1.17.122 Organization URL(s)

Location: /components/[]/licenses/[]/licensing/licensor/organization/url
Property: url (Optional)
Type: Array (of String)
Format: iri-reference as specified in RFC 3987

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

Example (Informative)
```
https://example.com
```


© Ecma International 2025

6.6.1.17.123 Organizational Contact

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

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.6.1.17.124 Organizational Person

Location: /components/[]/licenses/[]/licensing/licensor/organization/contact/[]
Type: Object

Table 66 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.17.125 BOM Reference

Location: /components/[]/licenses/[]/licensing/licensor/organization/contact/[]/bom-ref
Type: String

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

6.6.1.17.126 Name

Location: /components/[]/licenses/[]/licensing/licensor/organization/contact/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.6.1.17.127 Email Address

Location: /components/[]/licenses/[]/licensing/licensor/organization/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

© Ecma International 2025

6.6.1.17.128 Phone

Location: /components/[]/licenses/[]/licensing/licensor/organization/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)

```
800-555-1212
```

6.6.1.17.129 Licensor (Individual)

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

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

Table 67 — Properties for the individual object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.17.130 BOM Reference

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

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

6.6.1.17.131 Name

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

The name of a contact.

Example (Informative)

```
Contact name
```


© Ecma International 2025

6.6.1.17.132 Email Address

Location: /components/[]/licenses/[]/licensing/licensor/individual/email
Property: email (Optional)
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.6.1.17.133 Phone

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

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.6.1.17.134 Licensee

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

The individual or organization for which a licence was granted to.

Table 68 — Properties for the licensee object

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

6.6.1.17.135 Licensee (Organization)

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

The organization that was granted the licence.

© Ecma International 2025

Table 69 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.17.136 BOM Reference

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

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

6.6.1.17.137 Organization Name

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

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.6.1.17.138 Organization Address

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

The physical address (location) of the organization.

Table 70 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

© Ecma International 2025

Table 70 — Properties for the address object *(continued)*

Property	Type	Requirement	Description
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

6.6.1.17.139 BOM Reference

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

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

6.6.1.17.140 Country

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

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

6.6.1.17.141 Region

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

The region or state in the country.

Example (Informative)

```
Texas
```

6.6.1.17.142 Locality

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

The locality or city within the country.

Example (Informative)

```
Austin
```

© Ecma International 2025

6.6.1.17.143 Post Office Box Number

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

The post office box number.

Example (Informative)
```
901
```

6.6.1.17.144 Postal Code

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

The postal code.

Example (Informative)
```
78758
```

6.6.1.17.145 Street Address

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

The street address.

Example (Informative)
```
100 Main Street
```

6.6.1.17.146 Organization URL(s)

Location: /components/[]/licenses/[]/licensing/licensee/organization/url
Property: url (Optional)
Type: Array (of String)
Format: iri-reference as specified in RFC 3987

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

Example (Informative)
```
https://example.com
```

6.6.1.17.147 Organizational Contact

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

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

© Ecma International 2025

6.6.1.17.148 Organizational Person

Location: /components/[]/licenses/[]/licensing/licensee/organization/contact/[]
Type: Object

Table 71 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.17.149 BOM Reference

Location: /components/[]/licenses/[]/licensing/licensee/organization/contact/[]/bom-ref
Type: String

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

6.6.1.17.150 Name

Location: /components/[]/licenses/[]/licensing/licensee/organization/contact/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.6.1.17.151 Email Address

Location: /components/[]/licenses/[]/licensing/licensee/organization/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

© Ecma International 2025

6.6.1.17.152 Phone

Location: /components/[]/licenses/[]/licensing/licensee/organization/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)

```
800-555-1212
```

6.6.1.17.153 Licensee (Individual)

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

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

Table 72 — Properties for the individual object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.17.154 BOM Reference

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

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

6.6.1.17.155 Name

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

The name of a contact.

Example (Informative)

```
Contact name
```


© Ecma International 2025

6.6.1.17.156 Email Address

Location: /components/[]/licenses/[]/licensing/licensee/individual/email
Property: email (Optional)
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

6.6.1.17.157 Phone

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

The phone number of the contact.

Example (Informative)

```
800-555-1212
```

6.6.1.17.158 Purchaser

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

The individual or organization that purchased the licence.

Table 73 — Properties for the purchaser object

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

6.6.1.17.159 Purchaser (Organization)

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

The organization that purchased the licence.

© Ecma International 2025

Table 74 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.17.160 BOM Reference

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

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

6.6.1.17.161 Organization Name

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

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.6.1.17.162 Organization Address

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

The physical address (location) of the organization.

Table 75 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

© Ecma International 2025

Table 75 — Properties for the address object *(continued)*

Property	Type	Requirement	Description
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

6.6.1.17.163 BOM Reference

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

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

6.6.1.17.164 Country

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

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

6.6.1.17.165 Region

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

The region or state in the country.

Example (Informative)

```
Texas
```

6.6.1.17.166 Locality

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

The locality or city within the country.

Example (Informative)

```
Austin
```

© Ecma International 2025

6.6.1.17.167 Post Office Box Number

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

The post office box number.

Example (Informative)
```
901
```

6.6.1.17.168 Postal Code

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

The postal code.

Example (Informative)
```
78758
```

6.6.1.17.169 Street Address

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

The street address.

Example (Informative)
```
100 Main Street
```

6.6.1.17.170 Organization URL(s)

Location: /components/[]/licenses/[]/licensing/purchaser/organization/url
Property: url (Optional)
Type: Array (of String)
Format: iri-reference as specified in RFC 3987

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

Example (Informative)
```
https://example.com
```

6.6.1.17.171 Organizational Contact

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

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

© Ecma International 2025

6.6.1.17.172 Organizational Person

Location: /components/[]/licenses/[]/licensing/purchaser/organization/contact/[]
Type: Object

Table 76 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.17.173 BOM Reference

Location: /components/[]/licenses/[]/licensing/purchaser/organization/contact/[]/bom-ref
Type: String

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

6.6.1.17.174 Name

Location: /components/[]/licenses/[]/licensing/purchaser/organization/contact/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.6.1.17.175 Email Address

Location: /components/[]/licenses/[]/licensing/purchaser/organization/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

© Ecma International 2025

6.6.1.17.176 Phone

Location: /components/[]/licenses/[]/licensing/purchaser/organization/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)

```
800-555-1212
```

6.6.1.17.177 Purchaser (Individual)

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

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

Table 77 — Properties for the individual object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.17.178 BOM Reference

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

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

6.6.1.17.179 Name

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

The name of a contact.

Example (Informative)

```
Contact name
```


© Ecma International 2025

6.6.1.17.180 Email Address

Location: /components/[]/licenses/[]/licensing/purchaser/individual/email
Property: email (Optional)
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.6.1.17.181 Phone

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

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.6.1.17.182 Purchase Order

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

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

6.6.1.17.183 License Type

Location: /components/[]/licenses/[]/licensing/licenseTypes
Property: licenseTypes (Optional)
Type: Array (of String) (enum)

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

Table 78 — 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 licenses for a software application and licenses 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.

© Ecma International 2025

Table 78 — Enumeration of possible values *(continued)*

Value	Description
evaluation	A licence that grants permission to install 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.
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.

6.6.1.17.184 Last Renewal

Location: /components/[]/licenses/[]/licensing/lastRenewal
Property: lastRenewal (Optional)
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The timestamp indicating when the licence 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 licence was last renewed.

6.6.1.17.185 Expiration

Location: /components/[]/licenses/[]/licensing/expiration
Property: expiration (Optional)
Type: String
Format: data-time as specified in RFC 3339 section 5.6

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

6.6.1.17.186 Properties

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

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 shall be a Lightweight name-value pair object.

© Ecma International 2025

6.6.1.17.187 Lightweight name-value pair

Location: /components/[]/licenses/[]/properties/[]
Type: Object

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 79 — Properties for the property 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.

6.6.1.17.188 Name

Location: /components/[]/licenses/[]/properties/[]/name
Type: String

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

6.6.1.17.189 Value

Location: /components/[]/licenses/[]/properties/[]/value
Type: String

The value of the property.

6.6.1.18 Component Copyright

Location: /components/[]/copyright
Type: String

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

Example (Informative)

```
Acme Inc
```

6.6.1.19 Component Patent(s)

Location: /components/[]/patentAssertions
Property: patentAssertions (Optional)
Type: Array

A list of assertions made regarding patents associated with this component or service. Assertions distinguish between ownership, licensing, and other relevant interactions with patents. Each item of this array shall be a Patent Assertion object.

© Ecma International 2025

6.6.1.19.1 Patent Assertion

Location: /components/[]/patentAssertions/[]
Type: Object

An assertion linking a patent or patent family to this component or service.

Table 80 — Properties for the patentAssertion object

Property	Type	Requirement	Description
bom-ref	String	Optional	A reference to the patent or patent family object within the BOM. This shall match the **`bom-ref`** of a **`patent`** or **`patentFamily`** object.
assertionType	String	Required	The type of assertion being made about the patent or patent family. Examples include ownership, licensing, and standards inclusion.
patentRefs	Array	Optional	A list of BOM references (**`bom-ref`**) linking to patents or patent families associated with this assertion.
asserter	Array	Required	undefined.
notes	String	Optional	Additional notes or clarifications regarding the assertion, if necessary. For example, geographical restrictions, duration, or limitations of a licence.

6.6.1.19.2 BOM Reference

Location: /components/[]/patentAssertions/[]/bom-ref
Type: String

A reference to the patent or patent family object within the BOM. This shall match the **`bom-ref`** of a **`patent`** or **`patentFamily`** object.

6.6.1.19.3 Assertion Type

Location: /components/[]/patentAssertions/[]/assertionType
Type: String (enum)

The type of assertion being made about the patent or patent family. Examples include ownership, licensing, and standards inclusion.

Table 81 — Enumeration of possible values

Value	Description
ownership	The manufacturer asserts ownership of the patent or patent family.
license	The manufacturer asserts they have a licence to use the patent or patent family.
third-party-claim	A third party has asserted a claim or potential infringement against the manufacturer’s component or service.
standards-inclusion	The patent is part of a standard essential patent (SEP) portfolio relevant to the component or service.
prior-art	The manufacturer asserts the patent or patent family as prior art that invalidates another patent or claim.
exclusive-rights	The manufacturer asserts exclusive rights granted through a licensing agreement.

© Ecma International 2025

Table 81 — Enumeration of possible values *(continued)*

Value	Description
non-assertion	The manufacturer asserts they will not enforce the patent or patent family against certain uses or users.
research-or-evaluation	The patent or patent family is being used under a research or evaluation licence.

6.6.1.19.4 Patent References

Location: /components/[]/patentAssertions/[]/patentRefs
Property: patentRefs (Optional)
Type: Array (of String)

A list of BOM references (**`bom-ref`**) linking to patents or patent families associated with this assertion. 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 shall be a string.

6.6.1.19.5 Asserter

Location: /components/[]/patentAssertions/[]/asserter

Shall be one of:

1. Organizational Entity
2. Person
3. Reference

6.6.1.19.6 Organizational Entity

Type: Object

Table 82 — Properties for the organizational entity object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.19.7 Person

Type: Object

© Ecma International 2025

Table 83 — Properties for the person object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.19.8 Reference

Type: String

A reference to a previously defined **`organizationalContact`** or **`organizationalEntity`** object in the BOM. The value shall be a valid **`bom-ref`** pointing to one of these objects.

6.6.1.19.9 BOM Reference

Location: /components/[]/patentAssertions/[]/asserter/bom-ref
Property: bom-ref (Optional)
Type: String

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

6.6.1.19.10 Organization Name

Location: /components/[]/patentAssertions/[]/asserter/name
Property: name (Optional)
Type: String

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.6.1.19.11 Organization Address

Location: /components/[]/patentAssertions/[]/asserter/address
Property: address (Optional)
Type: Object

The physical address (location) of the organization.

© Ecma International 2025

Table 84 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.19.12 BOM Reference

Location: /components/[]/patentAssertions/[]/asserter/address/bom-ref
Property: bom-ref (Optional)
Type: String

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

6.6.1.19.13 Country

Location: /components/[]/patentAssertions/[]/asserter/address/country
Property: country (Optional)
Type: String

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

6.6.1.19.14 Region

Location: /components/[]/patentAssertions/[]/asserter/address/region
Property: region (Optional)
Type: String

The region or state in the country.

Example (Informative)

```
Texas
```

© Ecma International 2025

6.6.1.19.15 Locality

Location: /components/[]/patentAssertions/[]/asserter/address/locality
Property: locality (Optional)
Type: String

The locality or city within the country.

Example (Informative)
```
Austin
```

6.6.1.19.16 Post Office Box Number

Location: /components/[]/patentAssertions/[]/asserter/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)
Type: String

The post office box number.

Example (Informative)
```
901
```

6.6.1.19.17 Postal Code

Location: /components/[]/patentAssertions/[]/asserter/address/postalCode
Property: postalCode (Optional)
Type: String

The postal code.

Example (Informative)
```
78758
```

6.6.1.19.18 Street Address

Location: /components/[]/patentAssertions/[]/asserter/address/streetAddress
Property: streetAddress (Optional)
Type: String

The street address.

Example (Informative)
```
100 Main Street
```

6.6.1.19.19 Organization URL(s)

Location: /components/[]/patentAssertions/[]/asserter/url
Property: url (Optional)
Type: Array (of String)
Format: iri-reference as specified in RFC 3987

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

Example (Informative)
```
https://example.com
```


© Ecma International 2025

6.6.1.19.20 Organizational Contact

Location: /components/[]/patentAssertions/[]/asserter/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.6.1.19.21 Organizational Person

Location: /components/[]/patentAssertions/[]/asserter/contact/[]
Type: Object

Table 85 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.19.22 BOM Reference

Location: /components/[]/patentAssertions/[]/asserter/contact/[]/bom-ref
Type: String

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

6.6.1.19.23 Name

Location: /components/[]/patentAssertions/[]/asserter/contact/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.6.1.19.24 Email Address

Location: /components/[]/patentAssertions/[]/asserter/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

© Ecma International 2025

6.6.1.19.25 Phone

Location: /components/[]/patentAssertions/[]/asserter/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.6.1.19.26 BOM Reference

Location: /components/[]/patentAssertions/[]/asserter/bom-ref
Property: bom-ref (Optional)
Type: String

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

6.6.1.19.27 Name

Location: /components/[]/patentAssertions/[]/asserter/name
Property: name (Optional)
Type: String

The name of a contact.

Example (Informative)
```
Contact name
```

6.6.1.19.28 Email Address

Location: /components/[]/patentAssertions/[]/asserter/email
Property: email (Optional)
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.6.1.19.29 Phone

Location: /components/[]/patentAssertions/[]/asserter/phone
Property: phone (Optional)
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```


© Ecma International 2025

6.6.1.19.30 Notes

Location: /components/[]/patentAssertions/[]/notes
Type: String

Additional notes or clarifications regarding the assertion, if necessary. For example, geographical restrictions, duration, or limitations of a licence.

6.6.1.20 Common Platform Enumeration (CPE)

Location: /components/[]/cpe
Type: String

Asserts the identity of the component using CPE. The CPE shall 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.

Example (Informative)
```
cpe:2.3:a:acme:component_framework:-:*:*:*:*:*:*:*
```

6.6.1.21 Package URL (purl)

Location: /components/[]/purl
Type: String

Asserts the identity of the component using package-url (purl). The purl, if specified, shall 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.

Example (Informative)
```
pkg:maven/com.acme/tomcat-catalina@9.0.14?packaging=jar
```

6.6.1.22 OmniBOR Artifact Identifier (gitoid)

Location: /components/[]/omniborId
Property: omniborId (Optional)
Type: Array (of String)

Asserts the identity of the component using the OmniBOR Artifact ID. The OmniBOR, if specified, shall 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 shall be a string.

Example 1 (Informative)
```
gitoid:blob:sha1:a94a8fe5ccb19ba61c4c0873d391e987982fbbd3
```

Example 2 (Informative)
```
gitoid:blob:sha256:9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08
```

© Ecma International 2025

6.6.1.23 Software Heritage Identifier

Location: /components/[]/swhid
Property: swhid (Optional)
Type: Array (of String)

Asserts the identity of the component using the Software Heritage persistent identifier (SWHID). The SWHID, if specified, shall 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 shall be a string.

Example (Informative)

```
swh:1:cnt:94a9ed024d3859793618152ea559a168bbcbb5e2
```

6.6.1.24 SWID Tag

Location: /components/[]/swid
Type: Object

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 86 — 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.

6.6.1.24.1 Tag ID

Location: /components/[]/swid/tagId
Type: String

Maps to the tagId of a SoftwareIdentity.

6.6.1.24.2 Name

Location: /components/[]/swid/name
Type: String

Maps to the name of a SoftwareIdentity.

© Ecma International 2025

6.6.1.24.3 Version

Location: /components/[]/swid/version
Type: String
Default Value: `0.0`

Maps to the version of a SoftwareIdentity.

6.6.1.24.4 Tag Version

Location: /components/[]/swid/tagVersion
Type: Integer

Maps to the tagVersion of a SoftwareIdentity.

6.6.1.24.5 Patch

Location: /components/[]/swid/patch
Type: Boolean

Maps to the patch of a SoftwareIdentity.

6.6.1.24.6 Attachment text

Location: /components/[]/swid/text
Type: Object

Specifies the metadata and content of the SWID tag.

Table 87 — 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 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.

6.6.1.24.7 Content-Type

Location: /components/[]/swid/text/contentType
Type: String
Default Value: text/plain

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.

© Ecma International 2025

Example 1 (Informative)
```
text/plain
```

Example 2 (Informative)
```
application/json
```

Example 3 (Informative)
```
image/png
```

6.6.1.24.8 Encoding

Location: /components/[]/swid/text/encoding
Type: String (enum)

Specifies the encoding the text is represented in.

Table 88 — Enumeration of possible values

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

6.6.1.24.9 Attachment Text

Location: /components/[]/swid/text/content
Type: String

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

6.6.1.24.10 URL

Location: /components/[]/swid/url
Type: String
Format: iri-reference as specified in RFC 3987

The URL to the SWID file.

LEGACY

6.6.1.25 Component Modified From Original

Location: /components/[]/modified
Type: Boolean

[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.

© Ecma International 2025

6.6.1.26 Component Pedigree

Location: /components/[]/pedigree
Type: Object

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 89 — 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 an 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.

6.6.1.26.1 Ancestors

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

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 an 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 shall be a Component object.

6.6.1.26.2 Component

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

© Ecma International 2025

6.6.1.26.3 Descendants

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

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 shall be a Component object.

6.6.1.26.4 Component

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

6.6.1.26.5 Variants

Location: /components/[]/pedigree/variants
Property: variants (Optional)
Type: Array

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 shall be a Component object.

6.6.1.26.6 Component

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

6.6.1.26.7 Commits

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

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 shall be a Commit object.

6.6.1.26.8 Commit

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

Specifies an individual commit.

Table 90 — Properties for the commit 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.

© Ecma International 2025

Table 90 — Properties for the commit object *(continued)*

Property	Type	Requirement	Description
committer	Object	Optional	The person who committed or pushed the commit.
message	String	Optional	The text description of the contents of the commit.

6.6.1.26.9 UID

Location: /components/[]/pedigree/commits/[]/uid
Type: String

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

6.6.1.26.10 URL

Location: /components/[]/pedigree/commits/[]/url
Type: String
Format: iri-reference as specified in RFC 3987

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

6.6.1.26.11 Author

Location: /components/[]/pedigree/commits/[]/author
Type: Object

The author who created the changes in the commit.

Table 91 — 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.

6.6.1.26.12 Timestamp

Location: /components/[]/pedigree/commits/[]/author/timestamp
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The timestamp in which the action occurred.

6.6.1.26.13 Name

Location: /components/[]/pedigree/commits/[]/author/name
Type: String

The name of the individual who performed the action.

© Ecma International 2025

6.6.1.26.14 E-mail

Location: /components/[]/pedigree/commits/[]/author/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the individual who performed the action.

6.6.1.26.15 Committer

Location: /components/[]/pedigree/commits/[]/committer
Type: Object

The person who committed or pushed the commit.

Table 92 — 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.

6.6.1.26.16 Timestamp

Location: /components/[]/pedigree/commits/[]/committer/timestamp
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The timestamp in which the action occurred.

6.6.1.26.17 Name

Location: /components/[]/pedigree/commits/[]/committer/name
Type: String

The name of the individual who performed the action.

6.6.1.26.18 E-mail

Location: /components/[]/pedigree/commits/[]/committer/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the individual who performed the action.

6.6.1.26.19 Message

Location: /components/[]/pedigree/commits/[]/message
Type: String

The text description of the contents of the commit.

© Ecma International 2025

6.6.1.26.20 Patches

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

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 shall be a Patch object.

6.6.1.26.21 Patch

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

Specifies an individual patch.

Table 93 — Properties for the patch object

Property	Type	Requirement	Description
type	String	Required	Specifies the purpose for the patch including the resolution of defects, security issues, or new behaviour 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.

6.6.1.26.22 Patch Type

Location: /components/[]/pedigree/patches/[]/type
Type: String (enum)

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

Table 94 — 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.

6.6.1.26.23 Diff

Location: /components/[]/pedigree/patches/[]/diff
Type: Object

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

© Ecma International 2025

Table 95 — Properties for the diff object

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

6.6.1.26.24 Diff text

Location: /components/[]/pedigree/patches/[]/diff/text
Type: Object

Specifies the text of the diff.

Table 96 — 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 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.

6.6.1.26.25 Content-Type

Location: /components/[]/pedigree/patches/[]/diff/text/contentType
Type: String
Default Value: text/plain

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.

Example 1 (Informative)

```
text/plain
```

Example 2 (Informative)

```
application/json
```

Example 3 (Informative)

```
image/png
```

6.6.1.26.26 Encoding

Location: /components/[]/pedigree/patches/[]/diff/text/encoding
Type: String (enum)

Specifies the encoding the text is represented in.

© Ecma International 2025

Table 97 — Enumeration of possible values

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

6.6.1.26.27 Attachment Text

Location: /components/[]/pedigree/patches/[]/diff/text/content
Type: String

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

6.6.1.26.28 URL

Location: /components/[]/pedigree/patches/[]/diff/url
Type: String
Format: iri-reference as specified in RFC 3987

Specifies the URL to the diff.

6.6.1.26.29 Resolves

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

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

6.6.1.26.30 Issue

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

An individual issue that has been resolved.

Table 98 — Properties for the resolve 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.

6.6.1.26.31 Issue Type

Location: /components/[]/pedigree/patches/[]/resolves/[]/type
Type: String (enum)

Specifies the type of issue.

© Ecma International 2025

Table 99 — 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.

6.6.1.26.32 Issue ID

Location: /components/[]/pedigree/patches/[]/resolves/[]/id
Type: String

The identifier of the issue assigned by the source of the issue.

6.6.1.26.33 Issue Name

Location: /components/[]/pedigree/patches/[]/resolves/[]/name
Type: String

The name of the issue.

6.6.1.26.34 Issue Description

Location: /components/[]/pedigree/patches/[]/resolves/[]/description
Type: String

A description of the issue.

6.6.1.26.35 Source

Location: /components/[]/pedigree/patches/[]/resolves/[]/source
Type: Object

The source of the issue where it is documented.

Table 100 — 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.

6.6.1.26.36 Name

Location: /components/[]/pedigree/patches/[]/resolves/[]/source/name
Type: String

The name of the source.

Example 1 (Informative)

```
National Vulnerability Database
```


© Ecma International 2025

Example 2 (Informative)

```
NVD
```

Example 3 (Informative)

```
Apache
```

6.6.1.26.37 URL

Location: /components/[]/pedigree/patches/[]/resolves/[]/source/url
Type: String
Format: iri-reference as specified in RFC 3987

The url of the issue documentation as provided by the source.

6.6.1.26.38 References

Location: /components/[]/pedigree/patches/[]/resolves/[]/references
Property: references (Optional)
Type: Array (of String)
Format: iri-reference as specified in RFC 3987

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

Example (Informative)

```
https://example.com
```

6.6.1.26.39 Notes

Location: /components/[]/pedigree/notes
Type: String

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

6.6.1.27 External References

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

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 shall be an External Reference object.

6.6.1.27.1 External Reference

Location: /components/[]/externalReferences/[]
Type: Object

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.

© Ecma International 2025

6.6.1.28 Components

Location: /components/[]/components
Property: components (Optional)
Type: Array
Uniqueness: All items shall be unique.

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 shall be a Component object.

6.6.1.28.1 Component

Location: /components/[]/components/[]
Type: Object

6.6.1.29 Evidence

Location: /components/[]/evidence
Type: Object

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

Table 101 — 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	A list of SPDX licenses and/or named licenses and/or SPDX Licence Expression.
copyright	Array	Optional	Copyright evidence captures intellectual property assertions, providing evidence of possible ownership and legal protection.

6.6.1.29.1 Identity Evidence

Location: /components/[]/evidence/identity

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.

Shall be one of:

1. Array of Identity Objects
2. A Single Identity Object

© Ecma International 2025

6.6.1.29.2 Array of Identity Objects

Type: Array

LEGACY

6.6.1.29.3 A Single Identity Object

Type: Object

[Deprecated].

Table 102 — Properties for the a single identity object 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 analyse 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.

6.6.1.29.4 Array of Identity Objects

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

6.6.1.29.5 Identity Evidence

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

Evidence that substantiates the identity of a component.

Table 103 — 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).

© Ecma International 2025

Table 103 — Properties for the identity object *(continued)*

Property	Type	Requirement	Description
methods	Array	Optional	The methods used to extract and/or analyse 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.

6.6.1.29.6 Field

Location: /components/[]/evidence/identity/[]/field
Type: String (enum)

The identity field of the component which the evidence describes.

Enumeration of possible values:

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

6.6.1.29.7 Confidence

Location: /components/[]/evidence/identity/[]/confidence
Type: Number
Maximum Value: 1

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

6.6.1.29.8 Concluded Value

Location: /components/[]/evidence/identity/[]/concludedValue
Type: String

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

6.6.1.29.9 Methods

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

The methods used to extract and/or analyse the evidence.

6.6.1.29.10 Method

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

© Ecma International 2025

Table 104 — Properties for the method 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.

6.6.1.29.11 Technique

Location: /components/[]/evidence/identity/[]/methods/[]/technique
Type: String (enum)

The technique used in this method of analysis.

Enumeration of possible values:

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

6.6.1.29.12 Confidence

Location: /components/[]/evidence/identity/[]/methods/[]/confidence
Type: Number
Maximum Value: 1

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.

6.6.1.29.13 Value

Location: /components/[]/evidence/identity/[]/methods/[]/value
Type: String

The value or contents of the evidence.

6.6.1.29.14 BOM References

Location: /components/[]/evidence/identity/[]/tools
Property: tools (Optional)
Type: Array
Uniqueness: All items shall be unique.

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.

© Ecma International 2025

6.6.1.29.15 Tool

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

Shall be any of:

1. Ref
2. BOM-Link Element

6.6.1.29.16 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.6.1.29.17 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.6.1.29.18 Field

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

The identity field of the component which the evidence describes.

Enumeration of possible values:

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

6.6.1.29.19 Confidence

Location: /components/[]/evidence/identity/confidence
Property: confidence (Optional)
Type: Number
Maximum Value: 1

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

© Ecma International 2025

6.6.1.29.20 Concluded Value

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

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

6.6.1.29.21 Methods

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

The methods used to extract and/or analyse the evidence.

6.6.1.29.22 Method

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

Table 105 — Properties for the method 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.

6.6.1.29.23 Technique

Location: /components/[]/evidence/identity/methods/[]/technique
Type: String (enum)

The technique used in this method of analysis.

Enumeration of possible values:

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

© Ecma International 2025

6.6.1.29.24 Confidence

Location: /components/[]/evidence/identity/methods/[]/confidence
Type: Number
Maximum Value: 1

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.

6.6.1.29.25 Value

Location: /components/[]/evidence/identity/methods/[]/value
Type: String

The value or contents of the evidence.

6.6.1.29.26 BOM References

Location: /components/[]/evidence/identity/tools
Property: tools (Optional)
Type: Array
Uniqueness: All items shall be unique.

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.

6.6.1.29.27 Tool

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

Shall be any of:

1. Ref
2. BOM-Link Element

6.6.1.29.28 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.6.1.29.29 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

© Ecma International 2025

6.6.1.29.30 Occurrences

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

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

6.6.1.29.31 Occurrence

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

Table 106 — Properties for the occurrence object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the occurrence elsewhere in the BOM. Every **`bom-ref`** shall 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.
additionalContext	String	Optional	Any additional context of the detected component (e.g. a code snippet).

6.6.1.29.32 BOM Reference

Location: /components/[]/evidence/occurrences/[]/bom-ref
Type: String

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

6.6.1.29.33 Location

Location: /components/[]/evidence/occurrences/[]/location
Type: String

The location or path to where the component was found.

6.6.1.29.34 Line Number

Location: /components/[]/evidence/occurrences/[]/line
Type: Integer

The line number where the component was found.

© Ecma International 2025

6.6.1.29.35 Offset

Location: /components/[]/evidence/occurrences/[]/offset
Type: Integer

The offset where the component was found.

6.6.1.29.36 Symbol

Location: /components/[]/evidence/occurrences/[]/symbol
Type: String

The symbol name that was found associated with the component.

6.6.1.29.37 Additional Context

Location: /components/[]/evidence/occurrences/[]/additionalContext
Type: String

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

6.6.1.29.38 Call Stack

Location: /components/[]/evidence/callstack
Type: Object

Evidence of the components use through the callstack.

Table 107 — 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 programme execution and manages the sequence of function invocations.

6.6.1.29.39 Frames

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

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.

6.6.1.29.40 Frame

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

© Ecma International 2025

Table 108 — Properties for the frame 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	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.

6.6.1.29.41 Package

Location: /components/[]/evidence/callstack/frames/[]/package
Type: String

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

6.6.1.29.42 Module

Location: /components/[]/evidence/callstack/frames/[]/module
Type: String

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

6.6.1.29.43 Function

Location: /components/[]/evidence/callstack/frames/[]/function
Type: String

A block of code designed to perform a particular task.

6.6.1.29.44 Parameters

Location: /components/[]/evidence/callstack/frames/[]/parameters
Property: parameters (Optional)
Type: Array (of String)

Arguments that are passed to the module or function. Each item of this array shall be a string.

6.6.1.29.45 Line

Location: /components/[]/evidence/callstack/frames/[]/line
Type: Integer

The line number the code that is called resides on.

© Ecma International 2025

6.6.1.29.46 Column

Location: /components/[]/evidence/callstack/frames/[]/column
Type: Integer

The column the code that is called resides.

6.6.1.29.47 Full Filename

Location: /components/[]/evidence/callstack/frames/[]/fullFilename
Type: String

The full path and filename of the module.

6.6.1.29.48 License Evidence

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

A list of SPDX licenses and/or named licenses and/or SPDX Licence Expression.

6.6.1.29.49 License

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

6.6.1.29.50 Copyright Evidence

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

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

6.6.1.29.51 Copyright

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

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

Table 109 — Properties for the copyright object

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

6.6.1.29.52 Copyright Text

Location: /components/[]/evidence/copyright/[]/text
Type: String

The textual content of the copyright.

© Ecma International 2025

6.6.1.30 Release notes

Location: /components/[]/releaseNotes
Type: Object

Specifies release notes.

Table 110 — 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.

6.6.1.30.1 Type

Location: /components/[]/releaseNotes/type
Type: String

The software versioning type the release note describes.

Example 1 (Informative)

```
major
```

Example 2 (Informative)

```
minor
```

© Ecma International 2025

Example 3 (Informative)
```
patch
```

Example 4 (Informative)
```
pre-release
```

Example 5 (Informative)
```
internal
```

6.6.1.30.2 Title

Location: /components/[]/releaseNotes/title
Type: String

The title of the release.

6.6.1.30.3 Featured image

Location: /components/[]/releaseNotes/featuredImage
Type: String
Format: iri-reference as specified in RFC 3987

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

6.6.1.30.4 Social image

Location: /components/[]/releaseNotes/socialImage
Type: String
Format: iri-reference as specified in RFC 3987

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

6.6.1.30.5 Description

Location: /components/[]/releaseNotes/description
Type: String

A short description of the release.

6.6.1.30.6 Timestamp

Location: /components/[]/releaseNotes/timestamp
Type: String
Format: data-time as specified in RFC 3339 section 5.6

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

© Ecma International 2025

6.6.1.30.7 Aliases

Location: /components/[]/releaseNotes/aliases
Property: aliases (Optional)
Type: Array (of String)

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 shall be a string.

6.6.1.30.8 Tags

Location: /components/[]/releaseNotes/tags
Property: tags (Optional)
Type: Array (of String)

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 shall be a string.

Example 1 (Informative)
```
json-parser
```

Example 2 (Informative)
```
object-persistence
```

Example 3 (Informative)
```
text-to-image
```

Example 4 (Informative)
```
translation
```

Example 5 (Informative)
```
object-detection
```

6.6.1.30.9 Resolves

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

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

6.6.1.30.10 Issue

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

An individual issue that has been resolved.

© Ecma International 2025

Table 111 — Properties for the resolve 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.

6.6.1.30.11 Issue Type

Location: /components/[]/releaseNotes/resolves/[]/type
Type: String (enum)

Specifies the type of issue.

Table 112 — 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.

6.6.1.30.12 Issue ID

Location: /components/[]/releaseNotes/resolves/[]/id
Type: String

The identifier of the issue assigned by the source of the issue.

6.6.1.30.13 Issue Name

Location: /components/[]/releaseNotes/resolves/[]/name
Type: String

The name of the issue.

6.6.1.30.14 Issue Description

Location: /components/[]/releaseNotes/resolves/[]/description
Type: String

A description of the issue.

© Ecma International 2025

6.6.1.30.15 Source

Location: /components/[]/releaseNotes/resolves/[]/source
Type: Object

The source of the issue where it is documented.

Table 113 — 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.

6.6.1.30.16 Name

Location: /components/[]/releaseNotes/resolves/[]/source/name
Type: String

The name of the source.

Example 1 (Informative)
```
National Vulnerability Database
```

Example 2 (Informative)
```
NVD
```

Example 3 (Informative)
```
Apache
```

6.6.1.30.17 URL

Location: /components/[]/releaseNotes/resolves/[]/source/url
Type: String
Format: iri-reference as specified in RFC 3987

The url of the issue documentation as provided by the source.

6.6.1.30.18 References

Location: /components/[]/releaseNotes/resolves/[]/references
Property: references (Optional)
Type: Array (of String)
Format: iri-reference as specified in RFC 3987

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

Example (Informative)
```
https://example.com
```

© Ecma International 2025

6.6.1.30.19 Notes

Location: /components/[]/releaseNotes/notes
Property: notes (Optional)
Type: Array

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 shall be a Note object.

6.6.1.30.20 Note

Location: /components/[]/releaseNotes/notes/[]
Type: Object

A note containing the locale and content.

Table 114 — Properties for the note 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.

6.6.1.30.21 Locale

Location: /components/[]/releaseNotes/notes/[]/locale
Type: String
Pattern Constraint: `^([a-z]{2})(-[A-Z]{2})?$`

The ISO-639 (or higher) language code and optional ISO-3166 (or higher) country code. Examples include: "en", "en-US", "fr" and "fr-CA".

6.6.1.30.22 Release note content

Location: /components/[]/releaseNotes/notes/[]/text
Type: Object

Specifies the full content of the release note.

Table 115 — 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 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.

© Ecma International 2025

6.6.1.30.23 Content-Type

Location: /components/[]/releaseNotes/notes/[]/text/contentType
Type: String
Default Value: text/plain

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.

Example 1 (Informative)
```
text/plain
```

Example 2 (Informative)
```
application/json
```

Example 3 (Informative)
```
image/png
```

6.6.1.30.24 Encoding

Location: /components/[]/releaseNotes/notes/[]/text/encoding
Type: String (enum)

Specifies the encoding the text is represented in.

Table 116 — Enumeration of possible values

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

6.6.1.30.25 Attachment Text

Location: /components/[]/releaseNotes/notes/[]/text/content
Type: String

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

6.6.1.30.26 Properties

Location: /components/[]/releaseNotes/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

© Ecma International 2025

6.6.1.30.27 Lightweight name-value pair

Location: /components/[]/releaseNotes/properties/[]
Type: Object

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 117 — Properties for the property 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.

6.6.1.30.28 Name

Location: /components/[]/releaseNotes/properties/[]/name
Type: String

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

6.6.1.30.29 Value

Location: /components/[]/releaseNotes/properties/[]/value
Type: String

The value of the property.

6.6.1.31 AI/ML Model Card

Location: /components/[]/modelCard
Type: Object

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 shall not be specified for other component types.

Table 118 — Properties for the modelCard object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the model card elsewhere in the BOM. Every **`bom-ref`** shall 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.

© Ecma International 2025

Table 118 — Properties for the modelCard object *(continued)*

Property	Type	Requirement	Description
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.

6.6.1.31.1 BOM Reference

Location: /components/[]/modelCard/bom-ref
Type: String

An identifier which can be used to reference the model card elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. Value should not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

6.6.1.31.2 Model Parameters

Location: /components/[]/modelCard/modelParameters
Type: Object

Hyper-parameters for construction of the model.

Table 119 — 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.

© Ecma International 2025

6.6.1.31.3 Approach

Location: /components/[]/modelCard/modelParameters/approach
Type: Object

The overall approach to learning used by the model for problem solving.

Table 120 — Properties for the approach object

Property	Type	Requirement	Description
type	String	Optional	Learning types describing the learning problem or hybrid learning problem.

6.6.1.31.4 Learning Type

Location: /components/[]/modelCard/modelParameters/approach/type
Type: String (enum)

Learning types describing the learning problem or hybrid learning problem.

Table 121 — 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.

6.6.1.31.5 Task

Location: /components/[]/modelCard/modelParameters/task
Type: String

Directly influences the input and/or output. Examples include classification, regression, clustering, etc.

6.6.1.31.6 Architecture Family

Location: /components/[]/modelCard/modelParameters/architectureFamily
Type: String

The model architecture family such as transformer network, convolutional neural network, residual neural network, LSTM neural network, etc.

© Ecma International 2025

6.6.1.31.7 Model Architecture

Location: /components/[]/modelCard/modelParameters/modelArchitecture
Type: String

The specific architecture of the model such as GPT-1, ResNet-50, YOLOv3, etc.

6.6.1.31.8 Datasets

Location: /components/[]/modelCard/modelParameters/datasets
Property: datasets (Optional)
Type: Array

The datasets used to train and evaluate the model.

6.6.1.31.9 Dataset

Location: /components/[]/modelCard/modelParameters/datasets/[]

Shall be one of:

1. Inline Data Information
2. Data Reference

6.6.1.31.10 Inline Data Information

Type: Object

Table 122 — Properties for the inline data information object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the dataset elsewhere in the BOM. Every **`bom-ref`** shall 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.

© Ecma International 2025

6.6.1.31.11 Data Reference

Type: Object

Table 123 — Properties for the data reference object

Property	Type	Requirement	Description
ref	String	Optional	References a data component by the components bom-ref attribute.

6.6.1.31.12 BOM Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/bom-ref
Property: bom-ref (Optional)
Type: String

An identifier which can be used to reference the dataset elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. Value should not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

6.6.1.31.13 Type of Data

Location: /components/[]/modelCard/modelParameters/datasets/[]/type
Property: type (Required)
Type: String (enum)

The general theme or subject matter of the data being specified.

Table 124 — 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.

6.6.1.31.14 Dataset Name

Location: /components/[]/modelCard/modelParameters/datasets/[]/name
Property: name (Optional)
Type: String

The name of the dataset.

6.6.1.31.15 Data Contents

Location: /components/[]/modelCard/modelParameters/datasets/[]/contents
Property: contents (Optional)
Type: Object

The contents or references to the contents of the data being described.

© Ecma International 2025

Table 125 — Properties for the contents object

Property	Type	Requirement	Description
attachment	Object	Optional	A 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.

6.6.1.31.16 Data Attachment

Location: /components/[]/modelCard/modelParameters/datasets/[]/contents/attachment
Property: attachment (Optional)
Type: Object

A way to include textual or encoded data.

Table 126 — 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 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.

6.6.1.31.17 Content-Type

Location: /components/[]/modelCard/modelParameters/datasets/[]/contents/attachment/contentType
Property: contentType (Optional)
Type: String
Default Value: text/plain

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.

Example 1 (Informative)

```
text/plain
```

Example 2 (Informative)

```
application/json
```

Example 3 (Informative)

```
image/png
```

© Ecma International 2025

6.6.1.31.18 Encoding

Location: /components/[]/modelCard/modelParameters/datasets/[]/contents/attachment/encoding
Property: encoding (Optional)
Type: String (enum)

Specifies the encoding the text is represented in.

Table 127 — Enumeration of possible values

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

6.6.1.31.19 Attachment Text

Location: /components/[]/modelCard/modelParameters/datasets/[]/contents/attachment/content
Property: content (Required)
Type: String

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

6.6.1.31.20 Data URL

Location: /components/[]/modelCard/modelParameters/datasets/[]/contents/url
Property: url (Optional)
Type: String
Format: iri-reference as specified in RFC 3987

The URL to where the data can be retrieved.

6.6.1.31.21 Configuration Properties

Location: /components/[]/modelCard/modelParameters/datasets/[]/contents/properties
Property: properties (Optional)
Type: Array

Provides the ability to document name-value parameters used for configuration. Each item of this array shall be a Lightweight name-value pair object.

6.6.1.31.22 Lightweight name-value pair

Location: /components/[]/modelCard/modelParameters/datasets/[]/contents/properties/[]
Type: Object

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.

© Ecma International 2025

Table 128 — Properties for the property 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.

6.6.1.31.23 Name

Location: /components/[]/modelCard/modelParameters/datasets/[]/contents/properties/[]/name
Type: String

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

6.6.1.31.24 Value

Location: /components/[]/modelCard/modelParameters/datasets/[]/contents/properties/[]/value
Type: String

The value of the property.

6.6.1.31.25 Data Classification

Location: /components/[]/modelCard/modelParameters/datasets/[]/classification
Property: classification (Optional)
Type: String

Data classification tags data according to its type, sensitivity, and value if altered, stolen, or destroyed.

6.6.1.31.26 Sensitive Data

Location: /components/[]/modelCard/modelParameters/datasets/[]/sensitiveData
Property: sensitiveData (Optional)
Type: Array (of String)

A description of any sensitive data in a dataset. Each item of this array shall be a string.

6.6.1.31.27 Graphics Collection

Location: /components/[]/modelCard/modelParameters/datasets/[]/graphics
Property: graphics (Optional)
Type: Object

A collection of graphics that represent various measurements.

Table 129 — 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.

© Ecma International 2025

6.6.1.31.28 Description

Location: /components/[]/modelCard/modelParameters/datasets/[]/graphics/description
Property: description (Optional)
Type: String

A description of this collection of graphics.

6.6.1.31.29 Collection

Location: /components/[]/modelCard/modelParameters/datasets/[]/graphics/collection
Property: collection (Optional)
Type: Array

A collection of graphics. Each item of this array shall be a Graphic object.

6.6.1.31.30 Graphic

Location: /components/[]/modelCard/modelParameters/datasets/[]/graphics/collection/[]
Type: Object

Table 130 — 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 shall be specified for binary images.

6.6.1.31.31 Name

Location: /components/[]/modelCard/modelParameters/datasets/[]/graphics/collection/[]/name
Type: String

The name of the graphic.

6.6.1.31.32 Graphic Image

Location: /components/[]/modelCard/modelParameters/datasets/[]/graphics/collection/[]/image
Type: Object

The graphic (vector or raster). Base64 encoding shall be specified for binary images.

© Ecma International 2025

Table 131 — 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 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.

6.6.1.31.33 Content-Type

Location: /components/[]/modelCard/modelParameters/datasets/[]/graphics/collection/[]/image/contentType
Type: String
Default Value: text/plain

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.

Example 1 (Informative)
`text/plain`

Example 2 (Informative)
`application/json`

Example 3 (Informative)
`image/png`

6.6.1.31.34 Encoding

Location: /components/[]/modelCard/modelParameters/datasets/[]/graphics/collection/[]/image/encoding
Type: String (enum)

Specifies the encoding the text is represented in.

Table 132 — Enumeration of possible values

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

6.6.1.31.35 Attachment Text

Location: /components/[]/modelCard/modelParameters/datasets/[]/graphics/collection/[]/image/content
Type: String

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

© Ecma International 2025

6.6.1.31.36 Dataset Description

Location: /components/[]/modelCard/modelParameters/datasets/[]/description
Property: description (Optional)
Type: String

A description of the dataset. Can describe size of dataset, whether it's used for source code, training, testing, or validation, etc.

6.6.1.31.37 Data Governance

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance
Property: governance (Optional)
Type: Object

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 133 — 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.

6.6.1.31.38 Data Custodians

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians
Property: custodians (Optional)
Type: Array

Data custodians are responsible for the safe custody, transport, and storage of data.

6.6.1.31.39 Custodian

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]
Type: Object

Table 134 — Properties for the custodian 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).

© Ecma International 2025

6.6.1.31.40 Organization

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization
Type: Object

The organization that is responsible for specific data governance role(s).

Table 135 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.31.41 BOM Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/bom-ref
Type: String

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

6.6.1.31.42 Organization Name

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/name
Type: String

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.6.1.31.43 Organization Address

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/address
Type: Object

The physical address (location) of the organization.

© Ecma International 2025

Table 136 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.31.44 BOM Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/address/bom-ref
Type: String

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

6.6.1.31.45 Country

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/address/country
Type: String

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

6.6.1.31.46 Region

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/address/region
Type: String

The region or state in the country.

Example (Informative)

```
Texas
```


© Ecma International 2025

6.6.1.31.47 Locality

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/address/locality
Type: String

The locality or city within the country.

Example (Informative)

```
Austin
```

6.6.1.31.48 Post Office Box Number

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/address/postOfficeBoxNumber
Type: String

The post office box number.

Example (Informative)

```
901
```

6.6.1.31.49 Postal Code

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/address/postalCode
Type: String

The postal code.

Example (Informative)

```
78758
```

6.6.1.31.50 Street Address

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/address/streetAddress
Type: String

The street address.

Example (Informative)

```
100 Main Street
```

6.6.1.31.51 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

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

Example (Informative)

```
https://example.com
```

© Ecma International 2025

6.6.1.31.52 Organizational Contact

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.6.1.31.53 Organizational Person

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/contact/[]
Type: Object

Table 137 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.31.54 BOM Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/contact/[]/bom-ref
Type: String

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

6.6.1.31.55 Name

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/contact/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```


© Ecma International 2025

6.6.1.31.56 Email Address

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.6.1.31.57 Phone

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/organization/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.6.1.31.58 Individual

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/contact
Type: Object

The individual that is responsible for specific data governance role(s).

Table 138 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.31.59 BOM Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/contact/bom-ref
Type: String

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

© Ecma International 2025

6.6.1.31.60 Name

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/contact/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.6.1.31.61 Email Address

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/contact/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

6.6.1.31.62 Phone

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/custodians/[]/contact/phone
Type: String

The phone number of the contact.

Example (Informative)

```
800-555-1212
```

6.6.1.31.63 Data Stewards

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards
Property: stewards (Optional)
Type: Array

Data stewards are responsible for data content, context, and associated business rules.

6.6.1.31.64 Steward

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]
Type: Object

Table 139 — Properties for the steward 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).

© Ecma International 2025

6.6.1.31.65 Organization

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization
Type: Object

The organization that is responsible for specific data governance role(s).

Table 140 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.31.66 BOM Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/bom-ref
Type: String

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

6.6.1.31.67 Organization Name

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/name
Type: String

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.6.1.31.68 Organization Address

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/address
Type: Object

The physical address (location) of the organization.

© Ecma International 2025

Table 141 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.31.69 BOM Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/address/bom-ref
Type: String

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

6.6.1.31.70 Country

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/address/country
Type: String

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

6.6.1.31.71 Region

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/address/region
Type: String

The region or state in the country.

Example (Informative)

```
Texas
```


© Ecma International 2025

6.6.1.31.72 Locality

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/address/locality
Type: String

The locality or city within the country.

Example (Informative)

```
Austin
```

6.6.1.31.73 Post Office Box Number

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/address/postOfficeBoxNumber
Type: String

The post office box number.

Example (Informative)

```
901
```

6.6.1.31.74 Postal Code

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/address/postalCode
Type: String

The postal code.

Example (Informative)

```
78758
```

6.6.1.31.75 Street Address

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/address/streetAddress
Type: String

The street address.

Example (Informative)

```
100 Main Street
```

6.6.1.31.76 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

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

Example (Informative)

```
https://example.com
```

© Ecma International 2025

6.6.1.31.77 Organizational Contact

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.6.1.31.78 Organizational Person

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/contact/[]
Type: Object

Table 142 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.31.79 BOM Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/contact/[]/bom-ref
Type: String

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

6.6.1.31.80 Name

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/contact/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```


© Ecma International 2025

6.6.1.31.81 Email Address

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.6.1.31.82 Phone

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/organization/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.6.1.31.83 Individual

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/contact
Type: Object

The individual that is responsible for specific data governance role(s).

Table 143 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.31.84 BOM Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/contact/bom-ref
Type: String

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

© Ecma International 2025

6.6.1.31.85 Name

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/contact/name
Type: String

The name of a contact.

Example (Informative)
```
Contact name
```

6.6.1.31.86 Email Address

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/contact/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.6.1.31.87 Phone

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/stewards/[]/contact/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.6.1.31.88 Data Owners

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners
Property: owners (Optional)
Type: Array

Data owners are concerned with risk and appropriate access to data.

6.6.1.31.89 Owner

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]
Type: Object

Table 144 — Properties for the owner 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).

© Ecma International 2025

6.6.1.31.90 Organization

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization
Type: Object

The organization that is responsible for specific data governance role(s).

Table 145 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.31.91 BOM Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/bom-ref
Type: String

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

6.6.1.31.92 Organization Name

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/name
Type: String

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.6.1.31.93 Organization Address

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/address
Type: Object

The physical address (location) of the organization.

© Ecma International 2025

Table 146 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.31.94 BOM Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/address/bom-ref
Type: String

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

6.6.1.31.95 Country

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/address/country
Type: String

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

6.6.1.31.96 Region

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/address/region
Type: String

The region or state in the country.

Example (Informative)

```
Texas
```


© Ecma International 2025

6.6.1.31.97 Locality

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/address/locality
Type: String

The locality or city within the country.

Example (Informative)
```
Austin
```

6.6.1.31.98 Post Office Box Number

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/address/postOfficeBoxNumber
Type: String

The post office box number.

Example (Informative)
```
901
```

6.6.1.31.99 Postal Code

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/address/postalCode
Type: String

The postal code.

Example (Informative)
```
78758
```

6.6.1.31.100 Street Address

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/address/streetAddress
Type: String

The street address.

Example (Informative)
```
100 Main Street
```

6.6.1.31.101 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

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

Example (Informative)
```
https://example.com
```

© Ecma International 2025

6.6.1.31.102 Organizational Contact

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.6.1.31.103 Organizational Person

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/contact/[]
Type: Object

Table 147 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.31.104 BOM Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/contact/[]/bom-ref
Type: String

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

6.6.1.31.105 Name

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/contact/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```


© Ecma International 2025

6.6.1.31.106 Email Address

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
`firstname.lastname@example.com`

6.6.1.31.107 Phone

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/organization/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)
`800-555-1212`

6.6.1.31.108 Individual

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/contact
Type: Object

The individual that is responsible for specific data governance role(s).

Table 148 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.31.109 BOM Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/contact/bom-ref
Type: String

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

© Ecma International 2025

6.6.1.31.110 Name

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/contact/name
Type: String

The name of a contact.

Example (Informative)
```
Contact name
```

6.6.1.31.111 Email Address

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/contact/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.6.1.31.112 Phone

Location: /components/[]/modelCard/modelParameters/datasets/[]/governance/owners/[]/contact/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.6.1.31.113 Reference

Location: /components/[]/modelCard/modelParameters/datasets/[]/ref
Property: ref (Optional)
Type: String

References a data component by the components bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.6.1.31.114 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

© Ecma International 2025

6.6.1.31.115 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.6.1.31.116 Inputs

Location: /components/[]/modelCard/modelParameters/inputs
Property: inputs (Optional)
Type: Array

The input format(s) of the model Each item of this array shall be an Input and Output Parameters object.

6.6.1.31.117 Input and Output Parameters

Location: /components/[]/modelCard/modelParameters/inputs/[]
Type: Object

Table 149 — Properties for the input object

Property	Type	Requirement	Description
format	String	Optional	The data format for input/output to the model.

6.6.1.31.118 Input/Output Format

Location: /components/[]/modelCard/modelParameters/inputs/[]/format
Type: String

The data format for input/output to the model.

Example 1 (Informative)
```
string
```

Example 2 (Informative)
```
image
```

Example 3 (Informative)
```
time-series
```

6.6.1.31.119 Outputs

Location: /components/[]/modelCard/modelParameters/outputs
Property: outputs (Optional)
Type: Array

The output format(s) from the model Each item of this array shall be an Input and Output Parameters object.

© Ecma International 2025

6.6.1.31.120 Input and Output Parameters

Location: /components/[]/modelCard/modelParameters/outputs/[]
Type: Object

Table 150 — Properties for the output object

Property	Type	Requirement	Description
format	String	Optional	The data format for input/output to the model.

6.6.1.31.121 Input/Output Format

Location: /components/[]/modelCard/modelParameters/outputs/[]/format
Type: String

The data format for input/output to the model.

Example 1 (Informative)

```
string
```

Example 2 (Informative)

```
image
```

Example 3 (Informative)

```
time-series
```

6.6.1.31.122 Quantitative Analysis

Location: /components/[]/modelCard/quantitativeAnalysis
Type: Object

A quantitative analysis of the model.

Table 151 — 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.

6.6.1.31.123 Performance Metrics

Location: /components/[]/modelCard/quantitativeAnalysis/performanceMetrics
Property: performanceMetrics (Optional)
Type: Array

The model performance metrics being reported. Examples may include accuracy, F1 score, precision, top-3 error rates, MSC, etc. Each item of this array shall be a Performance Metric object.

© Ecma International 2025

6.6.1.31.124 Performance Metric

Location: /components/[]/modelCard/quantitativeAnalysis/performanceMetrics/[]
Type: Object

Table 152 — Properties for the performanceMetric 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.

6.6.1.31.125 Type

Location: /components/[]/modelCard/quantitativeAnalysis/performanceMetrics/[]/type
Type: String

The type of performance metric.

6.6.1.31.126 Value

Location: /components/[]/modelCard/quantitativeAnalysis/performanceMetrics/[]/value
Type: String

The value of the performance metric.

6.6.1.31.127 Slice

Location: /components/[]/modelCard/quantitativeAnalysis/performanceMetrics/[]/slice
Type: String

The name of the slice this metric was computed on. By default, assume this metric is not sliced.

6.6.1.31.128 Confidence Interval

Location: /components/[]/modelCard/quantitativeAnalysis/performanceMetrics/[]/confidenceInterval
Type: Object

The confidence interval of the metric.

Table 153 — 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.

© Ecma International 2025

6.6.1.31.129 Lower Bound

Location: /components/[]/modelCard/quantitativeAnalysis/performanceMetrics/[]/confidenceInterval/lowerBound
Type: String

The lower bound of the confidence interval.

6.6.1.31.130 Upper Bound

Location: /components/[]/modelCard/quantitativeAnalysis/performanceMetrics/[]/confidenceInterval/upperBound
Type: String

The upper bound of the confidence interval.

6.6.1.31.131 Graphics Collection

Location: /components/[]/modelCard/quantitativeAnalysis/graphics
Type: Object

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.

6.6.1.31.132 Description

Location: /components/[]/modelCard/quantitativeAnalysis/graphics/description
Type: String

A description of this collection of graphics.

6.6.1.31.133 Collection

Location: /components/[]/modelCard/quantitativeAnalysis/graphics/collection
Property: collection (Optional)
Type: Array

A collection of graphics. Each item of this array shall be a Graphic object.

6.6.1.31.134 Graphic

Location: /components/[]/modelCard/quantitativeAnalysis/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 shall be specified for binary images.

© Ecma International 2025

6.6.1.31.135 Name

Location: /components/[]/modelCard/quantitativeAnalysis/graphics/collection/[]/name
Type: String

The name of the graphic.

6.6.1.31.136 Graphic Image

Location: /components/[]/modelCard/quantitativeAnalysis/graphics/collection/[]/image
Type: Object

The graphic (vector or raster). Base64 encoding shall 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 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.

6.6.1.31.137 Content-Type

Location: /components/[]/modelCard/quantitativeAnalysis/graphics/collection/[]/image/contentType
Type: String
Default Value: text/plain

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.

Example 1 (Informative)

```
text/plain
```

Example 2 (Informative)

```
application/json
```

Example 3 (Informative)

```
image/png
```

6.6.1.31.138 Encoding

Location: /components/[]/modelCard/quantitativeAnalysis/graphics/collection/[]/image/encoding
Type: String (enum)

Specifies the encoding the text is represented in.

© Ecma International 2025

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.

6.6.1.31.139 Attachment Text

Location: /components/[]/modelCard/quantitativeAnalysis/graphics/collection/[]/image/content
Type: String

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

6.6.1.31.140 Considerations

Location: /components/[]/modelCard/considerations
Type: Object

What considerations should be taken into account regarding the model's construction, training, and application?.

Table 158 — 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?.

6.6.1.31.141 Users

Location: /components/[]/modelCard/considerations/users
Property: users (Optional)
Type: Array (of String)

Who are the intended users of the model? Each item of this array shall be a string.

© Ecma International 2025

6.6.1.31.142 Use Cases

Location: /components/[]/modelCard/considerations/useCases
Property: useCases (Optional)
Type: Array (of String)

What are the intended use cases of the model? Each item of this array shall be a string.

6.6.1.31.143 Technical Limitations

Location: /components/[]/modelCard/considerations/technicalLimitations
Property: technicalLimitations (Optional)
Type: Array (of String)

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 shall be a string.

6.6.1.31.144 Performance Tradeoffs

Location: /components/[]/modelCard/considerations/performanceTradeoffs
Property: performanceTradeoffs (Optional)
Type: Array (of String)

What are the known tradeoffs in accuracy/performance of the model? Each item of this array shall be a string.

6.6.1.31.145 Ethical Considerations

Location: /components/[]/modelCard/considerations/ethicalConsiderations
Property: ethicalConsiderations (Optional)
Type: Array

What are the ethical risks involved in the application of this model? Each item of this array shall be a Risk object.

6.6.1.31.146 Risk

Location: /components/[]/modelCard/considerations/ethicalConsiderations/[]
Type: Object

Table 159 — Properties for the ethicalConsideration object

Property	Type	Requirement	Description
name	String	Optional	The name of the risk.
mitigationStrategy	String	Optional	Strategy used to address this risk.

6.6.1.31.147 Name

Location: /components/[]/modelCard/considerations/ethicalConsiderations/[]/name
Type: String

The name of the risk.

© Ecma International 2025

6.6.1.31.148 Mitigation Strategy

Location: /components/[]/modelCard/considerations/ethicalConsiderations/[]/mitigationStrategy
Type: String

Strategy used to address this risk.

6.6.1.31.149 Environmental Considerations

Location: /components/[]/modelCard/considerations/environmentalConsiderations
Type: Object

What are the various environmental impacts the corresponding machine learning model has exhibited across its lifecycle?.

Table 160 — 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.

6.6.1.31.150 Energy Consumptions

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions
Property: energyConsumptions (Optional)
Type: Array

Describes energy consumption information incurred for one or more component lifecycle activities. Each item of this array shall be an Energy consumption object.

6.6.1.31.151 Energy consumption

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]
Type: Object

Describes energy consumption information incurred for the specified lifecycle activity.

Table 161 — Properties for the energyConsumption 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.

© Ecma International 2025

Table 161 — Properties for the energyConsumption object *(continued)*

Property	Type	Requirement	Description
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 general public are encouraged to be registered in the CycloneDX Property Taxonomy. Formal registration is optional.

6.6.1.31.152 Activity

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/activity
Type: String (enum)

The type of activity that is part of a machine learning model development or operational lifecycle.

Table 162 — 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.

6.6.1.31.153 Energy Providers

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders
Property: energyProviders (Required)
Type: Array

The provider(s) of the energy consumed by the associated model development lifecycle activity. Each item of this array shall be an Energy Provider object.

© Ecma International 2025

6.6.1.31.154 Energy Provider

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]
Type: Object

Describes the physical provider of energy used for model development or operations.

Table 163 — Properties for the energyProvider object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the energy provider elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.31.155 BOM Reference

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/bom-ref
Type: String

An identifier which can be used to reference the energy provider elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. Value should not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

6.6.1.31.156 Description

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/description
Type: String

A description of the energy provider.

6.6.1.31.157 Organization

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization
Type: Object

The organization that provides energy.

© Ecma International 2025

Table 164 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.31.158 BOM Reference

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/bom-ref
Type: String

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

6.6.1.31.159 Organization Name

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/name
Type: String

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.6.1.31.160 Organization Address

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/address
Type: Object

The physical address (location) of the organization.

Table 165 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

© Ecma International 2025

Table 165 — Properties for the address object *(continued)*

Property	Type	Requirement	Description
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

6.6.1.31.161 BOM Reference

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/address/bom-ref
Type: String

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

6.6.1.31.162 Country

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/address/country
Type: String

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

6.6.1.31.163 Region

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/address/region
Type: String

The region or state in the country.

Example (Informative)

```
Texas
```

6.6.1.31.164 Locality

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/address/locality
Type: String

The locality or city within the country.

Example (Informative)

```
Austin
```


© Ecma International 2025

6.6.1.31.165 Post Office Box Number

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/address/postOfficeBoxNumber
Type: String

The post office box number.

Example (Informative)
```
901
```

6.6.1.31.166 Postal Code

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/address/postalCode
Type: String

The postal code.

Example (Informative)
```
78758
```

6.6.1.31.167 Street Address

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/address/streetAddress
Type: String

The street address.

Example (Informative)
```
100 Main Street
```

6.6.1.31.168 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

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

Example (Informative)
```
https://example.com
```

6.6.1.31.169 Organizational Contact

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

© Ecma International 2025

6.6.1.31.170 Organizational Person

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/contact/[]
Type: Object

Table 166 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.31.171 BOM Reference

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/contact/[]/bom-ref
Type: String

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

6.6.1.31.172 Name

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/contact/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.6.1.31.173 Email Address

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```


© Ecma International 2025

6.6.1.31.174 Phone

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/organization/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)
800-555-1212

6.6.1.31.175 Energy Source

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/energySource
Type: String (enum)

The energy source for the energy provider.

Table 167 — 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 gases 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).
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.

6.6.1.31.176 Energy Provided

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/energyProvided
Type: Object

The energy provided by the energy source for an associated activity.

© Ecma International 2025

Table 168 — Properties for the energyProvided object

Property	Type	Requirement	Description
value	Number	Required	Quantity of energy.
unit	String	Required	Unit of energy.

6.6.1.31.177 Value

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/energyProvided/value
Type: Number

Quantity of energy.

6.6.1.31.178 Unit

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/energyProvided/unit
Type: String (enum)

Unit of energy.

Table 169 — Enumeration of possible values

Value	Description
kWh	Kilowatt-hour (kWh) is the energy delivered by one kilowatt (kW) of power for one hour (h).

6.6.1.31.179 External References

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/externalReferences
Property: externalReferences (Optional)
Type: Array

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 shall be an External Reference object.

6.6.1.31.180 External Reference

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/energyProviders/[]/externalReferences/[]
Type: Object

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.

6.6.1.31.181 Activity Energy Cost

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/activityEnergyCost
Type: Object

The total energy cost associated with the model lifecycle activity.

© Ecma International 2025

Table 170 — Properties for the activityEnergyCost object

Property	Type	Requirement	Description
value	Number	Required	Quantity of energy.
unit	String	Required	Unit of energy.

6.6.1.31.182 Value

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/activityEnergyCost/value
Type: Number

Quantity of energy.

6.6.1.31.183 Unit

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/activityEnergyCost/unit
Type: String (enum)

Unit of energy.

Table 171 — Enumeration of possible values

Value	Description
kWh	Kilowatt-hour (kWh) is the energy delivered by one kilowatt (kW) of power for one hour (h).

6.6.1.31.184 CO2 Equivalent Cost

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/co2CostEquivalent
Type: Object

The CO2 cost (debit) equivalent to the total energy cost.

Table 172 — 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).

6.6.1.31.185 Value

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/co2CostEquivalent/value
Type: Number

Quantity of carbon dioxide (CO2).

© Ecma International 2025

6.6.1.31.186 Unit

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/co2CostEquivalent/unit
Type: String (enum)

Unit of carbon dioxide (CO2).

Table 173 — Enumeration of possible values

Value	Description
tCO2eq	Tonnes (t) of carbon dioxide (CO2) equivalent (eq).

6.6.1.31.187 CO2 Cost Offset

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/co2CostOffset
Type: Object

The CO2 offset (credit) for the CO2 equivalent cost.

Table 174 — 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).

6.6.1.31.188 Value

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/co2CostOffset/value
Type: Number

Quantity of carbon dioxide (CO2).

6.6.1.31.189 Unit

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/co2CostOffset/unit
Type: String (enum)

Unit of carbon dioxide (CO2).

Table 175 — Enumeration of possible values

Value	Description
tCO2eq	Tonnes (t) of carbon dioxide (CO2) equivalent (eq).

© Ecma International 2025

6.6.1.31.190 Properties

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.6.1.31.191 Lightweight name-value pair

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/properties/[]
Type: Object

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 176 — Properties for the property 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.

6.6.1.31.192 Name

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/properties/[]/name
Type: String

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

6.6.1.31.193 Value

Location: /components/[]/modelCard/considerations/environmentalConsiderations/energyConsumptions/[]/properties/[]/value
Type: String

The value of the property.

6.6.1.31.194 Properties

Location: /components/[]/modelCard/considerations/environmentalConsiderations/properties
Property: properties (Optional)
Type: Array

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

© Ecma International 2025

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 shall be a Lightweight name-value pair object.

6.6.1.31.195 Lightweight name-value pair

Location: /components/[]/modelCard/considerations/environmentalConsiderations/properties/[]
Type: Object

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 177 — Properties for the property 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.

6.6.1.31.196 Name

Location: /components/[]/modelCard/considerations/environmentalConsiderations/properties/[]/name
Type: String

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

6.6.1.31.197 Value

Location: /components/[]/modelCard/considerations/environmentalConsiderations/properties/[]/value
Type: String

The value of the property.

6.6.1.31.198 Fairness Assessments

Location: /components/[]/modelCard/considerations/fairnessAssessments
Property: fairnessAssessments (Optional)
Type: Array

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 shall be a Fairness Assessment object.

6.6.1.31.199 Fairness Assessment

Location: /components/[]/modelCard/considerations/fairnessAssessments/[]
Type: Object

Information about the benefits and harms of the model to an identified at risk group.

© Ecma International 2025

Table 178 — Properties for the fairnessAssessment 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.

6.6.1.31.200 Group at Risk

Location: /components/[]/modelCard/considerations/fairnessAssessments/[]/groupAtRisk
Type: String

The groups or individuals at risk of being systematically disadvantaged by the model.

6.6.1.31.201 Benefits

Location: /components/[]/modelCard/considerations/fairnessAssessments/[]/benefits
Type: String

Expected benefits to the identified groups.

6.6.1.31.202 Harms

Location: /components/[]/modelCard/considerations/fairnessAssessments/[]/harms
Type: String

Expected harms to the identified groups.

6.6.1.31.203 Mitigation Strategy

Location: /components/[]/modelCard/considerations/fairnessAssessments/[]/mitigationStrategy
Type: String

With respect to the benefits and harms outlined, please describe any mitigation strategy implemented.

6.6.1.31.204 Properties

Location: /components/[]/modelCard/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

© Ecma International 2025

6.6.1.31.205 Lightweight name-value pair

Location: /components/[]/modelCard/properties/[]
Type: Object

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 179 — Properties for the property 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.

6.6.1.31.206 Name

Location: /components/[]/modelCard/properties/[]/name
Type: String

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

6.6.1.31.207 Value

Location: /components/[]/modelCard/properties/[]/value
Type: String

The value of the property.

6.6.1.32 Data

Location: /components/[]/data
Property: data (Optional)
Type: Array

This object should be specified for any component of type **`data`** and shall not be specified for other component types.

6.6.1.32.1 Data

Location: /components/[]/data/[]
Type: Object

© Ecma International 2025

Table 180 — Properties for the data object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the dataset elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.32.2 BOM Reference

Location: /components/[]/data/[]/bom-ref
Type: String

An identifier which can be used to reference the dataset elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. Value should not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

6.6.1.32.3 Type of Data

Location: /components/[]/data/[]/type
Type: String (enum)

The general theme or subject matter of the data being specified.

Table 181 — 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.

© Ecma International 2025

6.6.1.32.4 Dataset Name

Location: /components/[]/data/[]/name
Type: String

The name of the dataset.

6.6.1.32.5 Data Contents

Location: /components/[]/data/[]/contents
Type: Object

The contents or references to the contents of the data being described.

Table 182 — Properties for the contents object

Property	Type	Requirement	Description
attachment	Object	Optional	A 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.

6.6.1.32.6 Data Attachment

Location: /components/[]/data/[]/contents/attachment
Type: Object

A way to include textual or encoded data.

Table 183 — 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 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.

6.6.1.32.7 Content-Type

Location: /components/[]/data/[]/contents/attachment/contentType
Type: String
Default Value: text/plain

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.

© Ecma International 2025

Example 1 (Informative)
`text/plain`

Example 2 (Informative)
`application/json`

Example 3 (Informative)
`image/png`

6.6.1.32.8 Encoding

Location: /components/[]/data/[]/contents/attachment/encoding
Type: String (enum)

Specifies the encoding the text is represented in.

Table 184 — Enumeration of possible values

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

6.6.1.32.9 Attachment Text

Location: /components/[]/data/[]/contents/attachment/content
Type: String

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

6.6.1.32.10 Data URL

Location: /components/[]/data/[]/contents/url
Type: String
Format: iri-reference as specified in RFC 3987

The URL to where the data can be retrieved.

6.6.1.32.11 Configuration Properties

Location: /components/[]/data/[]/contents/properties
Property: properties (Optional)
Type: Array

Provides the ability to document name-value parameters used for configuration. Each item of this array shall be a Lightweight name-value pair object.

6.6.1.32.12 Lightweight name-value pair

Location: /components/[]/data/[]/contents/properties/[]
Type: Object

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

© Ecma International 2025

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

Table 185 — Properties for the property 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.

6.6.1.32.13 Name

Location: /components/[]/data/[]/contents/properties/[]/name
Type: String

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

6.6.1.32.14 Value

Location: /components/[]/data/[]/contents/properties/[]/value
Type: String

The value of the property.

6.6.1.32.15 Data Classification

Location: /components/[]/data/[]/classification
Type: String

Data classification tags data according to its type, sensitivity, and value if altered, stolen, or destroyed.

6.6.1.32.16 Sensitive Data

Location: /components/[]/data/[]/sensitiveData
Property: sensitiveData (Optional)
Type: Array (of String)

A description of any sensitive data in a dataset. Each item of this array shall be a string.

6.6.1.32.17 Graphics Collection

Location: /components/[]/data/[]/graphics
Type: Object

A collection of graphics that represent various measurements.

Table 186 — 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.

© Ecma International 2025

6.6.1.32.18 Description

Location: /components/[]/data/[]/graphics/description
Type: String

A description of this collection of graphics.

6.6.1.32.19 Collection

Location: /components/[]/data/[]/graphics/collection
Property: collection (Optional)
Type: Array

A collection of graphics. Each item of this array shall be a Graphic object.

6.6.1.32.20 Graphic

Location: /components/[]/data/[]/graphics/collection/[]
Type: Object

Table 187 — 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 shall be specified for binary images.

6.6.1.32.21 Name

Location: /components/[]/data/[]/graphics/collection/[]/name
Type: String

The name of the graphic.

6.6.1.32.22 Graphic Image

Location: /components/[]/data/[]/graphics/collection/[]/image
Type: Object

The graphic (vector or raster). Base64 encoding shall be specified for binary images.

Table 188 — 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 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.

© Ecma International 2025

6.6.1.32.23 Content-Type

Location: /components/[]/data/[]/graphics/collection/[]/image/contentType
Type: String
Default Value: text/plain

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.

Example 1 (Informative)
```
text/plain
```

Example 2 (Informative)
```
application/json
```

Example 3 (Informative)
```
image/png
```

6.6.1.32.24 Encoding

Location: /components/[]/data/[]/graphics/collection/[]/image/encoding
Type: String (enum)

Specifies the encoding the text is represented in.

Table 189 — Enumeration of possible values

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

6.6.1.32.25 Attachment Text

Location: /components/[]/data/[]/graphics/collection/[]/image/content
Type: String

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

6.6.1.32.26 Dataset Description

Location: /components/[]/data/[]/description
Type: String

A description of the dataset. Can describe size of dataset, whether it's used for source code, training, testing, or validation, etc.

© Ecma International 2025

6.6.1.32.27 Data Governance

Location: /components/[]/data/[]/governance
Type: Object

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 190 — 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.

6.6.1.32.28 Data Custodians

Location: /components/[]/data/[]/governance/custodians
Property: custodians (Optional)
Type: Array

Data custodians are responsible for the safe custody, transport, and storage of data.

6.6.1.32.29 Custodian

Location: /components/[]/data/[]/governance/custodians/[]
Type: Object

Table 191 — Properties for the custodian 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).

6.6.1.32.30 Organization

Location: /components/[]/data/[]/governance/custodians/[]/organization
Type: Object

The organization that is responsible for specific data governance role(s).

© Ecma International 2025

Table 192 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.32.31 BOM Reference

Location: /components/[]/data/[]/governance/custodians/[]/organization/bom-ref
Type: String

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

6.6.1.32.32 Organization Name

Location: /components/[]/data/[]/governance/custodians/[]/organization/name
Type: String

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.6.1.32.33 Organization Address

Location: /components/[]/data/[]/governance/custodians/[]/organization/address
Type: Object

The physical address (location) of the organization.

Table 193 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

© Ecma International 2025

Table 193 — Properties for the address object *(continued)*

Property	Type	Requirement	Description
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

6.6.1.32.34 BOM Reference

Location: /components/[]/data/[]/governance/custodians/[]/organization/address/bom-ref
Type: String

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

6.6.1.32.35 Country

Location: /components/[]/data/[]/governance/custodians/[]/organization/address/country
Type: String

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

6.6.1.32.36 Region

Location: /components/[]/data/[]/governance/custodians/[]/organization/address/region
Type: String

The region or state in the country.

Example (Informative)

```
Texas
```

6.6.1.32.37 Locality

Location: /components/[]/data/[]/governance/custodians/[]/organization/address/locality
Type: String

The locality or city within the country.

Example (Informative)

```
Austin
```

6.6.1.32.38 Post Office Box Number

Location: /components/[]/data/[]/governance/custodians/[]/organization/address/postOfficeBoxNumber
Type: String

The post office box number.

Example (Informative)

```
901
```

© Ecma International 2025

6.6.1.32.39 Postal Code

Location: /components/[]/data/[]/governance/custodians/[]/organization/address/postalCode
Type: String

The postal code.

Example (Informative)
```
78758
```

6.6.1.32.40 Street Address

Location: /components/[]/data/[]/governance/custodians/[]/organization/address/streetAddress
Type: String

The street address.

Example (Informative)
```
100 Main Street
```

6.6.1.32.41 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

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

Example (Informative)
```
https://example.com
```

6.6.1.32.42 Organizational Contact

Location: /components/[]/data/[]/governance/custodians/[]/organization/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.6.1.32.43 Organizational Person

Location: /components/[]/data/[]/governance/custodians/[]/organization/contact/[]
Type: Object

© Ecma International 2025

Table 194 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.32.44 BOM Reference

Location: /components/[]/data/[]/governance/custodians/[]/organization/contact/[]/bom-ref
Type: String

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

6.6.1.32.45 Name

Location: /components/[]/data/[]/governance/custodians/[]/organization/contact/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.6.1.32.46 Email Address

Location: /components/[]/data/[]/governance/custodians/[]/organization/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

6.6.1.32.47 Phone

Location: /components/[]/data/[]/governance/custodians/[]/organization/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)

```
800-555-1212
```

© Ecma International 2025

6.6.1.32.48 Individual

Location: /components/[]/data/[]/governance/custodians/[]/contact
Type: Object

The individual that is responsible for specific data governance role(s).

Table 195 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.32.49 BOM Reference

Location: /components/[]/data/[]/governance/custodians/[]/contact/bom-ref
Type: String

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

6.6.1.32.50 Name

Location: /components/[]/data/[]/governance/custodians/[]/contact/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.6.1.32.51 Email Address

Location: /components/[]/data/[]/governance/custodians/[]/contact/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```


© Ecma International 2025

6.6.1.32.52 Phone

Location: /components/[]/data/[]/governance/custodians/[]/contact/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.6.1.32.53 Data Stewards

Location: /components/[]/data/[]/governance/stewards
Property: stewards (Optional)
Type: Array

Data stewards are responsible for data content, context, and associated business rules.

6.6.1.32.54 Steward

Location: /components/[]/data/[]/governance/stewards/[]
Type: Object

Table 196 — Properties for the steward 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).

6.6.1.32.55 Organization

Location: /components/[]/data/[]/governance/stewards/[]/organization
Type: Object

The organization that is responsible for specific data governance role(s).

Table 197 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

© Ecma International 2025

6.6.1.32.56 BOM Reference

Location: /components/[]/data/[]/governance/stewards/[]/organization/bom-ref
Type: String

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

6.6.1.32.57 Organization Name

Location: /components/[]/data/[]/governance/stewards/[]/organization/name
Type: String

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.6.1.32.58 Organization Address

Location: /components/[]/data/[]/governance/stewards/[]/organization/address
Type: Object

The physical address (location) of the organization.

Table 198 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.32.59 BOM Reference

Location: /components/[]/data/[]/governance/stewards/[]/organization/address/bom-ref
Type: String

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

© Ecma International 2025

6.6.1.32.60 Country

Location: /components/[]/data/[]/governance/stewards/[]/organization/address/country
Type: String

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

6.6.1.32.61 Region

Location: /components/[]/data/[]/governance/stewards/[]/organization/address/region
Type: String

The region or state in the country.

Example (Informative)

```
Texas
```

6.6.1.32.62 Locality

Location: /components/[]/data/[]/governance/stewards/[]/organization/address/locality
Type: String

The locality or city within the country.

Example (Informative)

```
Austin
```

6.6.1.32.63 Post Office Box Number

Location: /components/[]/data/[]/governance/stewards/[]/organization/address/postOfficeBoxNumber
Type: String

The post office box number.

Example (Informative)

```
901
```

6.6.1.32.64 Postal Code

Location: /components/[]/data/[]/governance/stewards/[]/organization/address/postalCode
Type: String

The postal code.

Example (Informative)

```
78758
```

© Ecma International 2025

6.6.1.32.65 Street Address

Location: /components/[]/data/[]/governance/stewards/[]/organization/address/streetAddress
Type: String

The street address.

Example (Informative)

```
100 Main Street
```

6.6.1.32.66 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

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

Example (Informative)

```
https://example.com
```

6.6.1.32.67 Organizational Contact

Location: /components/[]/data/[]/governance/stewards/[]/organization/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.6.1.32.68 Organizational Person

Location: /components/[]/data/[]/governance/stewards/[]/organization/contact/[]
Type: Object

Table 199 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.32.69 BOM Reference

Location: /components/[]/data/[]/governance/stewards/[]/organization/contact/[]/bom-ref
Type: String

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

© Ecma International 2025

6.6.1.32.70 Name

Location: /components/[]/data/[]/governance/stewards/[]/organization/contact/[]/name
Type: String

The name of a contact.

Example (Informative)
```
Contact name
```

6.6.1.32.71 Email Address

Location: /components/[]/data/[]/governance/stewards/[]/organization/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.6.1.32.72 Phone

Location: /components/[]/data/[]/governance/stewards/[]/organization/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.6.1.32.73 Individual

Location: /components/[]/data/[]/governance/stewards/[]/contact
Type: Object

The individual that is responsible for specific data governance role(s).

Table 200 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

© Ecma International 2025

6.6.1.32.74 BOM Reference

Location: /components/[]/data/[]/governance/stewards/[]/contact/bom-ref
Type: String

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

6.6.1.32.75 Name

Location: /components/[]/data/[]/governance/stewards/[]/contact/name
Type: String

The name of a contact.

Example (Informative)
```
Contact name
```

6.6.1.32.76 Email Address

Location: /components/[]/data/[]/governance/stewards/[]/contact/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.6.1.32.77 Phone

Location: /components/[]/data/[]/governance/stewards/[]/contact/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.6.1.32.78 Data Owners

Location: /components/[]/data/[]/governance/owners
Property: owners (Optional)
Type: Array

Data owners are concerned with risk and appropriate access to data.

6.6.1.32.79 Owner

Location: /components/[]/data/[]/governance/owners/[]
Type: Object

© Ecma International 2025

Table 201 — Properties for the owner 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).

6.6.1.32.80 Organization

Location: /components/[]/data/[]/governance/owners/[]/organization
Type: Object

The organization that is responsible for specific data governance role(s).

Table 202 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.32.81 BOM Reference

Location: /components/[]/data/[]/governance/owners/[]/organization/bom-ref
Type: String

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

6.6.1.32.82 Organization Name

Location: /components/[]/data/[]/governance/owners/[]/organization/name
Type: String

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.6.1.32.83 Organization Address

Location: /components/[]/data/[]/governance/owners/[]/organization/address
Type: Object

The physical address (location) of the organization.

© Ecma International 2025

Table 203 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.32.84 BOM Reference

Location: /components/[]/data/[]/governance/owners/[]/organization/address/bom-ref
Type: String

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

6.6.1.32.85 Country

Location: /components/[]/data/[]/governance/owners/[]/organization/address/country
Type: String

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

6.6.1.32.86 Region

Location: /components/[]/data/[]/governance/owners/[]/organization/address/region
Type: String

The region or state in the country.

Example (Informative)

```
Texas
```

6.6.1.32.87 Locality

Location: /components/[]/data/[]/governance/owners/[]/organization/address/locality
Type: String

The locality or city within the country.

Example (Informative)

```
Austin
```


© Ecma International 2025

6.6.1.32.88 Post Office Box Number

Location: /components/[]/data/[]/governance/owners/[]/organization/address/postOfficeBoxNumber
Type: String

The post office box number.

Example (Informative)
```
901
```

6.6.1.32.89 Postal Code

Location: /components/[]/data/[]/governance/owners/[]/organization/address/postalCode
Type: String

The postal code.

Example (Informative)
```
78758
```

6.6.1.32.90 Street Address

Location: /components/[]/data/[]/governance/owners/[]/organization/address/streetAddress
Type: String

The street address.

Example (Informative)
```
100 Main Street
```

6.6.1.32.91 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

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

Example (Informative)
```
https://example.com
```

6.6.1.32.92 Organizational Contact

Location: /components/[]/data/[]/governance/owners/[]/organization/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.6.1.32.93 Organizational Person

Location: /components/[]/data/[]/governance/owners/[]/organization/contact/[]
Type: Object

© Ecma International 2025

Table 204 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.32.94 BOM Reference

Location: /components/[]/data/[]/governance/owners/[]/organization/contact/[]/bom-ref
Type: String

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

6.6.1.32.95 Name

Location: /components/[]/data/[]/governance/owners/[]/organization/contact/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.6.1.32.96 Email Address

Location: /components/[]/data/[]/governance/owners/[]/organization/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

6.6.1.32.97 Phone

Location: /components/[]/data/[]/governance/owners/[]/organization/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)

```
800-555-1212
```


© Ecma International 2025

6.6.1.32.98 Individual

Location: /components/[]/data/[]/governance/owners/[]/contact
Type: Object

The individual that is responsible for specific data governance role(s).

Table 205 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.6.1.32.99 BOM Reference

Location: /components/[]/data/[]/governance/owners/[]/contact/bom-ref
Type: String

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

6.6.1.32.100 Name

Location: /components/[]/data/[]/governance/owners/[]/contact/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.6.1.32.101 Email Address

Location: /components/[]/data/[]/governance/owners/[]/contact/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

© Ecma International 2025

6.6.1.32.102 Phone

Location: /components/[]/data/[]/governance/owners/[]/contact/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.6.1.33 Cryptographic Properties

Location: /components/[]/cryptoProperties
Type: Object

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 206 — 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.

6.6.1.33.1 Asset Type

Location: /components/[]/cryptoProperties/assetType
Type: String (enum)

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.

© Ecma International 2025

Table 207 — 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.

6.6.1.33.2 Algorithm Properties

Location: /components/[]/cryptoProperties/algorithmProperties
Type: Object

Additional properties specific to a cryptographic algorithm.

Table 208 — 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).
algorithmFamily	String	Optional	A valid algorithm family identifier. If specified, this value shall be one of the enumeration of valid algorithm Family identifiers defined in the **`cryptography-defs.schema.json`** subschema.
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).

© Ecma International 2025

Table 208 — Properties for the algorithmProperties object *(continued)*

Property	Type	Requirement	Description
curve	String	Optional	[Deprecated] This will be removed in a future version. Use **`@.ellipticCurve`** instead. 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.
ellipticCurve	String	Optional	The specific underlying Elliptic Curve (EC) definition employed which is an indicator of the level of security strength, performance and complexity. If specified, this value shall be one of the enumeration of valid elliptic curves identifiers defined in the **`cryptography-defs.schema.json`** subschema.
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.
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.

6.6.1.33.3 Primitive

Location: /components/[]/cryptoProperties/algorithmProperties/primitive
Type: String (enum)

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

© Ecma International 2025

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 209 — 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.
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.
key-wrap	Key-wrap is a cryptographic technique used to securely encrypt and protect cryptographic keys using algorithms like AES.
other	Another primitive type.
unknown	The primitive is not known.

© Ecma International 2025

6.6.1.33.4 Algorithm Family

Location: /components/[]/cryptoProperties/algorithmProperties/algorithmFamily
Type: String (enum)

A valid algorithm family identifier. If specified, this value shall be one of the enumeration of valid algorithm Family identifiers defined in the **`cryptography-defs.schema.json`** subschema.

Example 1 (Informative)
```
3DES
```

Example 2 (Informative)
```
Blowfish
```

Example 3 (Informative)
```
ECDH
```

Enumeration of possible values:

- 3DES
- 3GPP-XOR
- A5/1
- A5/2
- AES
- ARIA
- Ascon
- BLAKE2
- BLAKE3
- BLS
- Blowfish
- CAMELLIA
- CAST5
- CAST6
- CMAC
- CMEA
- ChaCha
- ChaCha20
- DES
- DSA
- ECDH
- ECDSA
- ECIES
- EdDSA
- ElGamal
- FFDH
- Fortuna
- GOST
- HC
- HKDF
- HMAC
- IDEA
- IKE-PRF
- KMAC
- LMS
- MD2
- MD4
- MD5
- MILENAGE

© Ecma International 2025

- ML-DSA
- ML-KEM
- MQV
- PBES1
- PBES2
- PBKDF1
- PBKDF2
- PBMAC1
- Poly1305
- RABBIT
- RC2
- RC4
- RC5
- RC6
- RIPEMD
- RSAES-OAEP
- RSAES-PKCS1
- RSASSA-PKCS1
- RSASSA-PSS
- SEED
- SHA-1
- SHA-2
- SHA-3
- SLH-DSA
- SNOW3G
- SP800-108
- Salsa20
- Serpent
- SipHash
- Skipjack
- TUAK
- Twofish
- Whirlpool
- X3DH
- XMSS
- Yarrow
- ZUC
- bcrypt

6.6.1.33.5 Parameter Set Identifier

Location: /components/[]/cryptoProperties/algorithmProperties/parameterSetIdentifier
Type: String

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).

LEGACY

6.6.1.33.6 Elliptic Curve

Location: /components/[]/cryptoProperties/algorithmProperties/curve
Type: String

[Deprecated] This will be removed in a future version. Use **`@.ellipticCurve`** instead. 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.

6.6.1.33.7 Elliptic Curve

Location: /components/[]/cryptoProperties/algorithmProperties/ellipticCurve
Type: String (enum)

The specific underlying Elliptic Curve (EC) definition employed which is an indicator of the level of security strength, performance and complexity. If specified, this value shall be one of the enumeration of valid elliptic curves identifiers defined in the **`cryptography-defs.schema.json`** subschema.

Enumeration of possible values:

- anssi/FRP256v1
- bls/BLS12-377
- bls/BLS12-381
- bls/BLS12-446
- bls/BLS12-455
- bls/BLS12-638
- bls/BLS24-477
- bls/Bandersnatch
- bn/bn158
- bn/bn190
- bn/bn222
- bn/bn254
- bn/bn286
- bn/bn318
- bn/bn350
- bn/bn382
- bn/bn414
- bn/bn446
- bn/bn478
- bn/bn510
- bn/bn542
- bn/bn574
- bn/bn606
- bn/bn638
- brainpool/brainpoolP160r1
- brainpool/brainpoolP160t1
- brainpool/brainpoolP192r1
- brainpool/brainpoolP192t1
- brainpool/brainpoolP224r1
- brainpool/brainpoolP224t1
- brainpool/brainpoolP256r1
- brainpool/brainpoolP256t1
- brainpool/brainpoolP320r1
- brainpool/brainpoolP320t1
- brainpool/brainpoolP384r1
- brainpool/brainpoolP384t1
- brainpool/brainpoolP512r1
- brainpool/brainpoolP512t1
- gost/gost256
- gost/gost512
- gost/id-GostR3410-2001-CryptoPro-A-ParamSet
- gost/id-GostR3410-2001-CryptoPro-B-ParamSet
- gost/id-GostR3410-2001-CryptoPro-C-ParamSet
- gost/id-tc26-gost-3410-12-512-paramSetA

© Ecma International 2025

- gost/id-tc26-gost-3410-12-512-paramSetB
- gost/id-tc26-gost-3410-2012-256-paramSetA
- gost/id-tc26-gost-3410-2012-512-paramSetC
- mnt/mnt1
- mnt/mnt2/1
- mnt/mnt2/2
- mnt/mnt3/1
- mnt/mnt3/2
- mnt/mnt3/3
- mnt/mnt4
- mnt/mnt5/1
- mnt/mnt5/2
- mnt/mnt5/3
- nist/B-163
- nist/B-233
- nist/B-283
- nist/B-409
- nist/B-571
- nist/K-163
- nist/K-233
- nist/K-283
- nist/K-409
- nist/K-571
- nist/P-192
- nist/P-224
- nist/P-256
- nist/P-384
- nist/P-521
- nums/ed-254-mont
- nums/ed-255-mers
- nums/ed-256-mont
- nums/ed-382-mont
- nums/ed-383-mers
- nums/ed-384-mont
- nums/ed-510-mont
- nums/ed-511-mers
- nums/ed-512-mont
- nums/numsp256d1
- nums/numsp256t1
- nums/numsp384d1
- nums/numsp384t1
- nums/numsp512d1
- nums/numsp512t1
- nums/w-254-mont
- nums/w-255-mers
- nums/w-256-mont
- nums/w-382-mont
- nums/w-383-mers
- nums/w-384-mont
- nums/w-510-mont
- nums/w-511-mers
- nums/w-512-mont
- oakley/192-bit Random ECP Group
- oakley/224-bit Random ECP Group
- oakley/256-bit Random ECP Group
- oakley/384-bit Random ECP Group
- oakley/521-bit Random ECP Group
- oakley/Oakley Group 3
- oakley/Oakley Group 4
- oscaa/SM2
- other/BADA55-R-256

© Ecma International 2025

- other/BADA55-VPR-224
- other/BADA55-VPR2-224
- other/BADA55-VR-224
- other/BADA55-VR-256
- other/BADA55-VR-384
- other/Curve1174
- other/Curve22103
- other/Curve25519
- other/Curve383187
- other/Curve41417
- other/Curve4417
- other/Curve448
- other/Curve67254
- other/E-222
- other/E-382
- other/E-521
- other/Ed25519
- other/Ed448
- other/Ed448-Goldilocks
- other/FourQ
- other/Fp224BN
- other/Fp254BNa
- other/Fp254BNb
- other/Fp254n2BNa
- other/Fp256BN
- other/Fp384BN
- other/Fp512BN
- other/JubJub
- other/M-221
- other/M-383
- other/M-511
- other/MDC201601
- other/Pallas
- other/Tom-256
- other/Tom-384
- other/Tom-521
- other/Tweedledee
- other/Tweedledum
- other/Vesta
- other/ssc-160
- other/ssc-192
- other/ssc-224
- other/ssc-256
- other/ssc-288
- other/ssc-320
- other/ssc-384
- other/ssc-512
- secg/secp112r1
- secg/secp112r2
- secg/secp128r1
- secg/secp128r2
- secg/secp160k1
- secg/secp160r1
- secg/secp160r2
- secg/secp192k1
- secg/secp192r1
- secg/secp224k1
- secg/secp224r1
- secg/secp256k1
- secg/secp256r1
- secg/secp384r1

© Ecma International 2025

- secg/secp521r1
- secg/sect113r1
- secg/sect113r2
- secg/sect131r1
- secg/sect131r2
- secg/sect163k1
- secg/sect163r1
- secg/sect163r2
- secg/sect193r1
- secg/sect193r2
- secg/sect233k1
- secg/sect233r1
- secg/sect239k1
- secg/sect283k1
- secg/sect283r1
- secg/sect409k1
- secg/sect409r1
- secg/sect571k1
- secg/sect571r1
- wtls/wap-wsg-idm-ecid-wtls1
- wtls/wap-wsg-idm-ecid-wtls10
- wtls/wap-wsg-idm-ecid-wtls11
- wtls/wap-wsg-idm-ecid-wtls12
- wtls/wap-wsg-idm-ecid-wtls3
- wtls/wap-wsg-idm-ecid-wtls4
- wtls/wap-wsg-idm-ecid-wtls5
- wtls/wap-wsg-idm-ecid-wtls6
- wtls/wap-wsg-idm-ecid-wtls7
- wtls/wap-wsg-idm-ecid-wtls8
- wtls/wap-wsg-idm-ecid-wtls9
- x962/c2onb191v4
- x962/c2onb191v5
- x962/c2onb239v4
- x962/c2onb239v5
- x962/c2pnb163v1
- x962/c2pnb163v2
- x962/c2pnb163v3
- x962/c2pnb176w1
- x962/c2pnb208w1
- x962/c2pnb272w1
- x962/c2pnb304w1
- x962/c2pnb368w1
- x962/c2tnb191v1
- x962/c2tnb191v2
- x962/c2tnb191v3
- x962/c2tnb239v1
- x962/c2tnb239v2
- x962/c2tnb239v3
- x962/c2tnb359v1
- x962/c2tnb431r1
- x962/prime192v1
- x962/prime192v2
- x962/prime192v3
- x962/prime239v1
- x962/prime239v2
- x962/prime239v3
- x962/prime256v1
- x963/ansip160k1
- x963/ansip160r1
- x963/ansip160r2
- x963/ansip192k1

© Ecma International 2025

- x963/ansip224k1
- x963/ansip224r1
- x963/ansip256k1
- x963/ansip384r1
- x963/ansip521r1
- x963/ansit163k1
- x963/ansit163r1
- x963/ansit163r2
- x963/ansit193r1
- x963/ansit193r2
- x963/ansit233k1
- x963/ansit233r1
- x963/ansit239k1
- x963/ansit283k1
- x963/ansit283r1
- x963/ansit409k1
- x963/ansit409r1
- x963/ansit571k1
- x963/ansit571r1

6.6.1.33.8 Execution Environment

Location: /components/[]/cryptoProperties/algorithmProperties/executionEnvironment
Type: String (enum)

The target and execution environment in which the algorithm is implemented in.

Table 210 — 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.

6.6.1.33.9 Implementation platform

Location: /components/[]/cryptoProperties/algorithmProperties/implementationPlatform
Type: String (enum)

The target platform for which the algorithm is implemented. The implementation can be 'generic', running on any platform or for a specific platform.

Enumeration of possible values:

- generic
- x86_32
- x86_64
- armv7-a
- armv7-m
- armv8-a
- armv8-m
- armv9-a

© Ecma International 2025

- armv9-m
- s390x
- ppc64
- ppc64le
- other
- unknown

6.6.1.33.10 Certification Level

Location: /components/[]/cryptoProperties/algorithmProperties/certificationLevel
Property: certificationLevel (Optional)
Type: Array (of String) (enum)

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 shall be a string.

Table 211 — 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.

© Ecma International 2025

Table 211 — Enumeration of possible values *(continued)*

Value	Description
cc-eal5+	Common Criteria - Evaluation Assurance Level 5 (Augmented).
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.

6.6.1.33.11 Mode

Location: /components/[]/cryptoProperties/algorithmProperties/mode
Type: String (enum)

The mode of operation in which the cryptographic algorithm (block cipher) is used.

Table 212 — 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.

6.6.1.33.12 Padding

Location: /components/[]/cryptoProperties/algorithmProperties/padding
Type: String (enum)

The padding scheme that is used for the cryptographic algorithm.

© Ecma International 2025

Table 213 — 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.

6.6.1.33.13 Cryptographic functions

Location: /components/[]/cryptoProperties/algorithmProperties/cryptoFunctions
Property: cryptoFunctions (Optional)
Type: Array (of String) (enum)

The cryptographic functions implemented by the cryptographic algorithm. Each item of this array shall be a string.

Enumeration of possible values:

- generate
- keygen
- encrypt
- decrypt
- digest
- tag
- keyderive
- sign
- verify
- encapsulate
- decapsulate
- other
- unknown

6.6.1.33.14 Classical security level

Location: /components/[]/cryptoProperties/algorithmProperties/classicalSecurityLevel
Type: Integer

The classical security level that a cryptographic algorithm provides (in bits).

6.6.1.33.15 NIST security strength category

Location: /components/[]/cryptoProperties/algorithmProperties/nistQuantumSecurityLevel
Type: Integer
Maximum Value: 6

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.

© Ecma International 2025

6.6.1.33.16 Certificate Properties

Location: /components/[]/cryptoProperties/certificateProperties
Type: Object

Properties for cryptographic assets of asset type 'certificate'.

Table 214 — Properties for the certificateProperties object

Property	Type	Requirement	Description
serialNumber	String	Optional	The serial number is a unique identifier for the certificate issued by a CA.
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	[DEPRECATED] This will be removed in a future version. Use **`@.relatedCryptographicAssets`** instead. The bom-ref to signature algorithm used by the certificate.
subjectPublicKeyRef	String	Optional	[DEPRECATED] This will be removed in a future version. Use **`@.relatedCryptographicAssets`** instead. The bom-ref to the public key of the subject.
certificateFormat	String	Optional	The format of the certificate.
certificateExtension	String	Optional	[DEPRECATED] This will be removed in a future version. Use **`@.certificateFileExtension`** instead. The file extension of the certificate.
certificateFileExtension	String	Optional	The file extension of the certificate.
fingerprint	Object	Optional	The fingerprint is a cryptographic hash of the certificate excluding it's signature.
certificateState	Array	Optional	The certificate lifecycle is a comprehensive process that manages digital certificates from their initial creation to eventual expiration or revocation. It typically involves several stages.
creationDate	String	Optional	The date and time (timestamp) when the certificate was created or pre-activated.
activationDate	String	Optional	The date and time (timestamp) when the certificate was activated.
deactivationDate	String	Optional	The date and time (timestamp) when the related certificate was deactivated.
revocationDate	String	Optional	The date and time (timestamp) when the certificate was revoked.
destructionDate	String	Optional	The date and time (timestamp) when the certificate was destroyed.

© Ecma International 2025

Table 214 — Properties for the certificateProperties object *(continued)*

Property	Type	Requirement	Description
certificateExtensions	Array	Optional	A certificate extension is a field that provides additional information about the certificate or its use. Extensions are used to convey additional information beyond the standard fields.
relatedCryptographicAssets	Array	Optional	A list of cryptographic assets related to this component.

6.6.1.33.17 Serial Number

Location: /components/[]/cryptoProperties/certificateProperties/serialNumber
Type: String

The serial number is a unique identifier for the certificate issued by a CA.

6.6.1.33.18 Subject Name

Location: /components/[]/cryptoProperties/certificateProperties/subjectName
Type: String

The subject name for the certificate.

6.6.1.33.19 Issuer Name

Location: /components/[]/cryptoProperties/certificateProperties/issuerName
Type: String

The issuer name for the certificate.

6.6.1.33.20 Not Valid Before

Location: /components/[]/cryptoProperties/certificateProperties/notValidBefore
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time according to ISO-8601 standard from which the certificate is valid.

© Ecma International 2025

6.6.1.33.21 Not Valid After

Location: /components/[]/cryptoProperties/certificateProperties/notValidAfter
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time according to ISO-8601 standard from which the certificate is not valid anymore.

LEGACY

6.6.1.33.22 Algorithm Reference

Location: /components/[]/cryptoProperties/certificateProperties/signatureAlgorithmRef
Type: String

[DEPRECATED] This will be removed in a future version. Use **`@.relatedCryptographicAssets`** instead. The bom-ref to signature algorithm used by the certificate.

6.6.1.33.23 Key reference

Location: /components/[]/cryptoProperties/certificateProperties/subjectPublicKeyRef
Type: String

[DEPRECATED] This will be removed in a future version. Use **`@.relatedCryptographicAssets`** instead. The bom-ref to the public key of the subject.

6.6.1.33.24 Certificate Format

Location: /components/[]/cryptoProperties/certificateProperties/certificateFormat
Type: String

The format of the certificate.

Example 1 (Informative)
```
X.509
```

Example 2 (Informative)
```
PEM
```

Example 3 (Informative)
```
DER
```


© Ecma International 2025

Example 4 (Informative)
```
CVC
```

LEGACY

6.6.1.33.25 Certificate File Extension

Location: /components/[]/cryptoProperties/certificateProperties/certificateExtension
Type: String

[DEPRECATED] This will be removed in a future version. Use **`@.certificateFileExtension`** instead. The file extension of the certificate.

Example 1 (Informative)
```
crt
```

Example 2 (Informative)
```
pem
```

Example 3 (Informative)
```
cer
```

Example 4 (Informative)
```
der
```

Example 5 (Informative)
```
p12
```

6.6.1.33.26 Certificate File Extension

Location: /components/[]/cryptoProperties/certificateProperties/certificateFileExtension
Type: String

The file extension of the certificate.

Example 1 (Informative)
```
crt
```

Example 2 (Informative)
```
pem
```

Example 3 (Informative)
```
cer
```

Example 4 (Informative)
```
der
```

Example 5 (Informative)
```
p12
```

© Ecma International 2025

6.6.1.33.27 Certificate Fingerprint

Location: /components/[]/cryptoProperties/certificateProperties/fingerprint
Type: Object

The fingerprint is a cryptographic hash of the certificate excluding it's signature.

Table 215 — Properties for the fingerprint object

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

6.6.1.33.28 Hash Algorithm

Location: /components/[]/cryptoProperties/certificateProperties/fingerprint/alg
Type: String (enum)

The algorithm that generated the hash value.

Enumeration of possible values:

- MD5
- SHA-1
- SHA-256
- SHA-384
- SHA-512
- SHA3-256
- SHA3-384
- SHA3-512
- BLAKE2b-256
- BLAKE2b-384
- BLAKE2b-512
- BLAKE3
- Streebog-256
- Streebog-512

6.6.1.33.29 Hash Value

Location: /components/[]/cryptoProperties/certificateProperties/fingerprint/content
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})$`

The value of the hash.

Example (Informative)
`3942447fac867ae5cdb3229b658f4d48`

© Ecma International 2025

6.6.1.33.30 Certificate Lifecycle State

Location: /components/[]/cryptoProperties/certificateProperties/certificateState
Property: certificateState (Optional)
Type: Array

The certificate lifecycle is a comprehensive process that manages digital certificates from their initial creation to eventual expiration or revocation. It typically involves several stages Each item of this array shall be a State object.

6.6.1.33.31 State

Location: /components/[]/cryptoProperties/certificateProperties/certificateState/[]
Type: Object

The state of the certificate.

Shall be one of:

1. Pre-Defined State
2. Custom State

6.6.1.33.32 Pre-Defined State

Type: Object

Table 216 — Properties for the pre-defined state object

Property	Type	Requirement	Description
state	String	Required	A pre-defined state in the certificate lifecycle.
reason	String	Optional	A reason for the certificate being in this state.

6.6.1.33.33 Custom State

Type: Object

Table 217 — Properties for the custom state object

Property	Type	Requirement	Description
name	String	Required	The name of the certificate lifecycle state.
description	String	Optional	The description of the certificate lifecycle state.
reason	String	Optional	A reason for the certificate being in this state.

6.6.1.33.34 State

Location: /components/[]/cryptoProperties/certificateProperties/certificateState/[]/state
Property: state (Required)
Type: String (enum)

A pre-defined state in the certificate lifecycle.

© Ecma International 2025

Table 218 — Enumeration of possible values

Value	Description
pre-activation	The certificate has been issued by the issuing certificate authority (CA) but has not been authorized for use.
active	The certificate may be used to cryptographically protect information, cryptographically process previously protected information, or both.
deactivated	Certificates in the deactivated state shall not be used to apply cryptographic protection but, in some cases, may be used to process cryptographically protected information.
suspended	The use of a certificate may be suspended for several possible reasons.
revoked	A revoked certificate is a digital certificate that has been invalidated by the issuing certificate authority (CA) before its scheduled expiration date.
destroyed	The certificate has been destroyed.

6.6.1.33.35 Reason

Location: /components/[]/cryptoProperties/certificateProperties/certificateState/[]/reason
Property: reason (Optional)
Type: String

A reason for the certificate being in this state.

6.6.1.33.36 State

Location: /components/[]/cryptoProperties/certificateProperties/certificateState/[]/name
Property: name (Required)
Type: String

The name of the certificate lifecycle state.

6.6.1.33.37 Description

Location: /components/[]/cryptoProperties/certificateProperties/certificateState/[]/description
Property: description (Optional)
Type: String

The description of the certificate lifecycle state.

6.6.1.33.38 Reason

Location: /components/[]/cryptoProperties/certificateProperties/certificateState/[]/reason
Property: reason (Optional)
Type: String

A reason for the certificate being in this state.

6.6.1.33.39 Creation Date

Location: /components/[]/cryptoProperties/certificateProperties/creationDate
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the certificate was created or pre-activated.

© Ecma International 2025

6.6.1.33.40 Activation Date

Location: /components/[]/cryptoProperties/certificateProperties/activationDate
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the certificate was activated.

6.6.1.33.41 Deactivation Date

Location: /components/[]/cryptoProperties/certificateProperties/deactivationDate
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the related certificate was deactivated.

6.6.1.33.42 Revocation Date

Location: /components/[]/cryptoProperties/certificateProperties/revocationDate
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the certificate was revoked.

6.6.1.33.43 Destruction Date

Location: /components/[]/cryptoProperties/certificateProperties/destructionDate
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the certificate was destroyed.

6.6.1.33.44 Certificate Extensions

Location: /components/[]/cryptoProperties/certificateProperties/certificateExtensions
Property: certificateExtensions (Optional)
Type: Array

A certificate extension is a field that provides additional information about the certificate or its use. Extensions are used to convey additional information beyond the standard fields. Each item of this array shall be an Extension object.

6.6.1.33.45 Extension

Location: /components/[]/cryptoProperties/certificateProperties/certificateExtensions/[]
Type: Object

© Ecma International 2025

Shall be one of:

1. Common Extensions
2. Custom Extensions

6.6.1.33.46 Common Extensions

Type: Object

Table 219 — Properties for the common extensions object

Property	Type	Requirement	Description
commonExtensionName	String	Required	The name of the extension.
commonExtensionValue	String	Required	The value of the certificate extension.

6.6.1.33.47 Custom Extensions

Type: Object

Custom extensions may convey application-specific or vendor-specific data not covered by standard extensions. The structure and semantics of custom extensions are typically defined outside of public standards. CycloneDX leverages properties to support this capability.

Table 220 — Properties for the custom extensions object

Property	Type	Requirement	Description
customExtensionName	String	Required	The name for the custom certificate extension.
customExtensionValue	String	Optional	The description of the custom certificate extension.

6.6.1.33.48 Name

Location: /components/[]/cryptoProperties/certificateProperties/certificateExtensions/[]/commonExtensionName
Property: commonExtensionName (Required)
Type: String (enum)

The name of the extension.

Table 221 — Enumeration of possible values

Value	Description
basicConstraints	Specifies whether a certificate can be used as a CA certificate or not.
keyUsage	Specifies the allowed uses of the public key in the certificate.
extendedKeyUsage	Specifies additional purposes for which the public key can be used.
subjectAlternativeName	Allows inclusion of additional names to identify the entity associated with the certificate.
authorityKeyIdentifier	Identifies the public key of the CA that issued the certificate.
subjectKeyIdentifier	Identifies the public key associated with the entity the certificate was issued to.

© Ecma International 2025

Table 221 — Enumeration of possible values *(continued)*

Value	Description
authorityInformationAccess	Contains CA issuers and OCSP information.
certificatePolicies	Defines the policies under which the certificate was issued and can be used.
crlDistributionPoints	Contains one or more URLs where a Certificate Revocation List (CRL) can be obtained.
signedCertificateTimestamp	Shows that the certificate has been publicly logged, which helps prevent the issuance of rogue certificates by a CA. Log ID, timestamp and signature as proof.

6.6.1.33.49 Value

Location: /components/[]/cryptoProperties/certificateProperties/certificateExtensions/[]/commonExtensionValue
Property: commonExtensionValue (Required)
Type: String

The value of the certificate extension.

6.6.1.33.50 Name

Location: /components/[]/cryptoProperties/certificateProperties/certificateExtensions/[]/customExtensionName
Property: customExtensionName (Required)
Type: String

The name for the custom certificate extension.

6.6.1.33.51 Value

Location: /components/[]/cryptoProperties/certificateProperties/certificateExtensions/[]/customExtensionValue
Property: customExtensionValue (Optional)
Type: String

The description of the custom certificate extension.

6.6.1.33.52 Related Cryptographic Assets

Location: /components/[]/cryptoProperties/certificateProperties/relatedCryptographicAssets
Property: relatedCryptographicAssets (Optional)
Type: Array

A list of cryptographic assets related to this component. Each item of this array shall be a Related Cryptographic Asset object.

6.6.1.33.53 Related Cryptographic Asset

Location: /components/[]/cryptoProperties/certificateProperties/relatedCryptographicAssets/[]
Type: Object

A cryptographic assets related to this component.

© Ecma International 2025

Table 222 — Properties for the relatedCryptographicAsset object

Property	Type	Requirement	Description
type	String	Optional	Specifies the mechanism by which the cryptographic asset is secured by.
ref	String	Optional	The bom-ref to cryptographic asset.

6.6.1.33.54 Type

Location: /components/[]/cryptoProperties/certificateProperties/relatedCryptographicAssets/[]/type
Type: String

Specifies the mechanism by which the cryptographic asset is secured by.

Example 1 (Informative)
```
publicKey
```

Example 2 (Informative)
```
privateKey
```

Example 3 (Informative)
```
algorithm
```

6.6.1.33.55 Reference to cryptographic asset

Location: /components/[]/cryptoProperties/certificateProperties/relatedCryptographicAssets/[]/ref
Type: String

The bom-ref to cryptographic asset.

6.6.1.33.56 Related Cryptographic Material Properties

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties
Type: Object

Properties for cryptographic assets of asset type: **`related-crypto-material`**.

Table 223 — Properties for the relatedCryptoMaterialProperties object

Property	Type	Requirement	Description
type	String	Optional	The type for the related cryptographic material.
id	String	Optional	The unique identifier for the related cryptographic material.
state	String	Optional	The key state as defined by NIST SP 800-57.
algorithmRef	String	Optional	[DEPRECATED] Use **`@.relatedCryptographicAssets`** instead. 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.

© Ecma International 2025

Table 223 — Properties for the relatedCryptoMaterialProperties object *(continued)*

Property	Type	Requirement	Description
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.
fingerprint	Object	Optional	The fingerprint is a cryptographic hash of the asset.
relatedCryptographicAssets	Array	Optional	A list of cryptographic assets related to this component.

6.6.1.33.57 RelatedCryptoMaterialType

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/type
Type: String (enum)

The type for the related cryptographic material.

Table 224 — 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.

© Ecma International 2025

Table 224 — Enumeration of possible values *(continued)*

Value	Description
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.

6.6.1.33.58 ID

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/id
Type: String

The unique identifier for the related cryptographic material.

6.6.1.33.59 State

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/state
Type: String (enum)

The key state as defined by NIST SP 800-57.

Enumeration of possible values:

- pre-activation
- active
- suspended
- deactivated
- compromised
- destroyed

LEGACY

6.6.1.33.60 Algorithm Reference

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/algorithmRef
Type: String

[DEPRECATED] Use **`@.relatedCryptographicAssets`** instead. The bom-ref to the algorithm used to generate the related cryptographic material.

© Ecma International 2025

6.6.1.33.61 Creation Date

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/creationDate
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the related cryptographic material was created.

6.6.1.33.62 Activation Date

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/activationDate
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the related cryptographic material was activated.

6.6.1.33.63 Update Date

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/updateDate
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the related cryptographic material was updated.

6.6.1.33.64 Expiration Date

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/expirationDate
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the related cryptographic material expires.

6.6.1.33.65 Value

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/value
Type: String

The associated value of the cryptographic material.

6.6.1.33.66 Size

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/size
Type: Integer

The size of the cryptographic asset (in bits).

6.6.1.33.67 Format

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/format
Type: String

The format of the related cryptographic material (e.g. P8, PEM, DER).

© Ecma International 2025

6.6.1.33.68 Secured By

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/securedBy
Type: Object

The mechanism by which the cryptographic asset is secured by.

Table 225 — 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.

6.6.1.33.69 Mechanism

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/securedBy/mechanism
Type: String

Specifies the mechanism by which the cryptographic asset is secured by.

Example 1 (Informative)
```
HSM
```

Example 2 (Informative)
```
TPM
```

Example 3 (Informative)
```
SGX
```

Example 4 (Informative)
```
Software
```

Example 5 (Informative)
```
None
```

6.6.1.33.70 Algorithm Reference

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/securedBy/algorithmRef
Type: String

The bom-ref to the algorithm.

6.6.1.33.71 Fingerprint

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/fingerprint
Type: Object

The fingerprint is a cryptographic hash of the asset.

© Ecma International 2025

Table 226 — Properties for the fingerprint object

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

6.6.1.33.72 Hash Algorithm

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/fingerprint/alg
Type: String (enum)

The algorithm that generated the hash value.

Enumeration of possible values:

- MD5
- SHA-1
- SHA-256
- SHA-384
- SHA-512
- SHA3-256
- SHA3-384
- SHA3-512
- BLAKE2b-256
- BLAKE2b-384
- BLAKE2b-512
- BLAKE3
- Streebog-256
- Streebog-512

6.6.1.33.73 Hash Value

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/fingerprint/content
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})$`

The value of the hash.

Example (Informative)
`3942447fac867ae5cdb3229b658f4d48`

6.6.1.33.74 Related Cryptographic Assets

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/relatedCryptographicAssets
Property: relatedCryptographicAssets (Optional)
Type: Array

A list of cryptographic assets related to this component. Each item of this array shall be a Related Cryptographic Asset object.

© Ecma International 2025

6.6.1.33.75 Related Cryptographic Asset

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/relatedCryptographicAssets/[]
Type: Object

A cryptographic assets related to this component.

Table 227 — Properties for the relatedCryptographicAsset object

Property	Type	Requirement	Description
type	String	Optional	Specifies the mechanism by which the cryptographic asset is secured by.
ref	String	Optional	The bom-ref to cryptographic asset.

6.6.1.33.76 Type

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/relatedCryptographicAssets/[]/type
Type: String

Specifies the mechanism by which the cryptographic asset is secured by.

Example 1 (Informative)
```
publicKey
```

Example 2 (Informative)
```
privateKey
```

Example 3 (Informative)
```
algorithm
```

6.6.1.33.77 Reference to cryptographic asset

Location: /components/[]/cryptoProperties/relatedCryptoMaterialProperties/relatedCryptographicAssets/[]/ref
Type: String

The bom-ref to cryptographic asset.

6.6.1.33.78 Protocol Properties

Location: /components/[]/cryptoProperties/protocolProperties
Type: Object

Properties specific to cryptographic assets of type: **`protocol`**.

Table 228 — 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.

© Ecma International 2025

Table 228 — Properties for the protocolProperties object *(continued)*

Property	Type	Requirement	Description
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	[DEPRECATED] Use **`@.relatedCryptographicAssets`** instead. A list of protocol-related cryptographic assets.
relatedCryptographicAssets	Array	Optional	A list of cryptographic assets related to this component.

6.6.1.33.79 Type

Location: /components/[]/cryptoProperties/protocolProperties/type
Type: String (enum)

The concrete protocol type.

Table 229 — Enumeration of possible values

Value	Description
tls	Transport Layer Security.
ssh	Secure Shell.
ipsec	Internet Protocol Security.
ike	Internet Key Exchange.
sstp	Secure Socket Tunneling Protocol.
wpa	Wi-Fi Protected Access.
dtls	Datagram Transport Layer Security.
quic	Quick UDP Internet Connections.
eap-aka	Extensible Authentication Protocol variant.
eap-aka-prime	Enhanced version of EAP-AKA.
prins	Protection of Inter-Network Signaling.
5g-aka	Authentication and Key Agreement for 5G.
other	Another protocol type.
unknown	The protocol type is not known.

6.6.1.33.80 Protocol Version

Location: /components/[]/cryptoProperties/protocolProperties/version
Type: String

The version of the protocol.

Example 1 (Informative)

```
1.0
```

© Ecma International 2025

Example 2 (Informative)

```
1.2
```

Example 3 (Informative)

```
1.99
```

6.6.1.33.81 Cipher Suites

Location: /components/[]/cryptoProperties/protocolProperties/cipherSuites
Property: cipherSuites (Optional)
Type: Array

A list of cipher suites related to the protocol. Each item of this array shall be a Cipher Suite object.

6.6.1.33.82 Cipher Suite

Location: /components/[]/cryptoProperties/protocolProperties/cipherSuites/[]
Type: Object

Object representing a cipher suite.

Table 230 — Properties for the cipherSuite 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.
tlsGroups	Array	Optional	A list of TLS named groups (formerly known as curves) for this cipher suite. These groups define the parameters for key exchange algorithms like ECDHE.
tlsSignatureSchemes	Array	Optional	A list of signature schemes supported for cipher suite. These schemes specify the algorithms used for digital signatures in TLS handshakes and certificate verification.

6.6.1.33.83 Common Name

Location: /components/[]/cryptoProperties/protocolProperties/cipherSuites/[]/name
Type: String

A common name for the cipher suite.

Example (Informative)

```
TLS_DHE_RSA_WITH_AES_128_CCM
```

6.6.1.33.84 Related Algorithms

Location: /components/[]/cryptoProperties/protocolProperties/cipherSuites/[]/algorithms
Property: algorithms (Optional)
Type: Array (of String)

A list of algorithms related to the cipher suite. The bom-ref to algorithm cryptographic asset. Each item of this array shall be a string.

© Ecma International 2025

6.6.1.33.85 Cipher Suite Identifiers

Location: /components/[]/cryptoProperties/protocolProperties/cipherSuites/[]/identifiers
Property: identifiers (Optional)
Type: Array (of String)

A list of common identifiers for the cipher suite. Cipher suite identifier Each item of this array shall be a string.

Example 1 (Informative)
```
0xC0
```

Example 2 (Informative)
```
0x9E
```

6.6.1.33.86 TLS Groups

Location: /components/[]/cryptoProperties/protocolProperties/cipherSuites/[]/tlsGroups
Property: tlsGroups (Optional)
Type: Array (of String)

A list of TLS named groups (formerly known as curves) for this cipher suite. These groups define the parameters for key exchange algorithms like ECDHE. The name of the TLS group Each item of this array shall be a string.

Example 1 (Informative)
```
x25519
```

Example 2 (Informative)
```
ffdhe2048
```

6.6.1.33.87 TLS Signature Schemes

Location: /components/[]/cryptoProperties/protocolProperties/cipherSuites/[]/tlsSignatureSchemes
Property: tlsSignatureSchemes (Optional)
Type: Array (of String)

A list of signature schemes supported for cipher suite. These schemes specify the algorithms used for digital signatures in TLS handshakes and certificate verification. The name of the TLS signature scheme Each item of this array shall be a string.

Example 1 (Informative)
```
ecdsa_secp256r1_sha256
```

Example 2 (Informative)
```
rsa_pss_rsae_sha256
```

Example 3 (Informative)
```
ed25519
```

6.6.1.33.88 IKEv2 Transform Types

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes
Type: Object

The IKEv2 transform types supported (types 1-4), defined in RFC 7296 section 3.3.2, and additional properties.

© Ecma International 2025

Table 231 — 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 per RFC9593.

6.6.1.33.89 Encryption Algorithms (ENCR)

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/encr

Transform Type 1: encryption algorithms.

Shall be any of:

1. Encryption Algorithms (ENCR)
2. Encryption Algorithm (ENCR) References

6.6.1.33.90 Encryption Algorithms (ENCR)

Type: Array

LEGACY

6.6.1.33.91 Encryption Algorithm (ENCR) References

Type: Array

[DEPRECATED] This will be removed in a future version. Transform Type 1: encryption algorithms.

6.6.1.33.92 Encryption Algorithms (ENCR)

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/encr
Property: encr
Type: Array

6.6.1.33.93 Encryption Algorithm (ENCR)

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/encr/[]
Type: Object

Object representing an encryption algorithm (ENCR).

© Ecma International 2025

Table 232 — Properties for the encr object

Property	Type	Requirement	Description
name	String	Optional	A name for the encryption method.
keyLength	Integer	Optional	The key length of the encryption algorithm.
algorithm	String	Optional	The bom-ref to algorithm cryptographic asset.

6.6.1.33.94 Name

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/encr/[]/name
Type: String

A name for the encryption method.

Example (Informative)
`ENCR_AES_GCM_16`

6.6.1.33.95 Encryption algorithm key length

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/encr/[]/keyLength
Type: Integer

The key length of the encryption algorithm.

6.6.1.33.96 Algorithm reference

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/encr/[]/algorithm
Type: String

The bom-ref to algorithm cryptographic asset.

LEGACY

6.6.1.33.97 Encryption Algorithm (ENCR) References

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/encr
Property: encr
Type: Array (of String)

[DEPRECATED] This will be removed in a future version. 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 shall be a string.

6.6.1.33.98 Pseudorandom Functions (PRF)

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/prf

Transform Type 2: pseudorandom functions.

Shall be any of:

1. Pseudorandom Functions (PRF)
2. Encryption Algorithm (ENCR) Reference Array

© Ecma International 2025

6.6.1.33.99 Pseudorandom Functions (PRF)

Type: Array

LEGACY

6.6.1.33.100 Encryption Algorithm (ENCR) Reference Array

Type: Array

[DEPRECATED] This will be removed in a future version. Transform Type 2: pseudorandom functions.

6.6.1.33.101 Pseudorandom Functions (PRF)

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/prf
Property: prf
Type: Array

6.6.1.33.102 Pseudorandom Function (PRF)

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/prf/[]
Type: Object

Object representing a pseudorandom function (PRF).

Table 233 — Properties for the prf object

Property	Type	Requirement	Description
name	String	Optional	A name for the pseudorandom function.
algorithm	String	Optional	The bom-ref to algorithm cryptographic asset.

6.6.1.33.103 Name

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/prf/[]/name
Type: String

A name for the pseudorandom function.

Example (Informative)
`PRF_HMAC_SHA2_256`

© Ecma International 2025

6.6.1.33.104 Algorithm reference

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/prf/[]/algorithm
Type: String

The bom-ref to algorithm cryptographic asset.

LEGACY

6.6.1.33.105 Encryption Algorithm (ENCR) Reference Array

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/prf
Property: prf
Type: Array (of String)

[DEPRECATED] This will be removed in a future version. 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 shall be a string.

6.6.1.33.106 Integrity Algorithms (INTEG)

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/integ

Transform Type 3: integrity algorithms.

Shall be any of:

1. Integrity Algorithms (INTEG)
2. Encryption Algorithm (ENCR) Reference Array

6.6.1.33.107 Integrity Algorithms (INTEG)

Type: Array

LEGACY

6.6.1.33.108 Encryption Algorithm (ENCR) Reference Array

Type: Array

[DEPRECATED] This will be removed in a future version. Transform Type 3: integrity algorithms.

6.6.1.33.109 Integrity Algorithms (INTEG)

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/integ
Property: integ
Type: Array

6.6.1.33.110 Integrity Algorithm (INTEG)

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/integ/[]
Type: Object

Object representing an integrity algorithm (INTEG).

© Ecma International 2025

Table 234 — Properties for the integ object

Property	Type	Requirement	Description
name	String	Optional	A name for the integrity algorithm.
algorithm	String	Optional	The bom-ref to algorithm cryptographic asset.

6.6.1.33.111 Name

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/integ/[]/name
Type: String

A name for the integrity algorithm.

Example (Informative)
```
AUTH_HMAC_SHA2_256_128
```

6.6.1.33.112 Algorithm reference

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/integ/[]/algorithm
Type: String

The bom-ref to algorithm cryptographic asset.

LEGACY

6.6.1.33.113 Encryption Algorithm (ENCR) Reference Array

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/integ
Property: integ
Type: Array (of String)

[DEPRECATED] This will be removed in a future version. 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 shall be a string.

6.6.1.33.114 Key Exchange Methods (KE)

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/ke

Transform Type 4: Key Exchange Method (KE) per RFC 9370, formerly called Diffie-Hellman Group (D-H).

Shall be any of:

1. Key Exchange Methods (KE)
2. Encryption Algorithm (ENCR) Reference Array

© Ecma International 2025

6.6.1.33.115 Key Exchange Methods (KE)

Type: Array

LEGACY

6.6.1.33.116 Encryption Algorithm (ENCR) Reference Array

Type: Array

[DEPRECATED] This will be removed in a future version. Transform Type 4: Key Exchange Method (KE) per RFC 9370, formerly called Diffie-Hellman Group (D-H).

6.6.1.33.117 Key Exchange Methods (KE)

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/ke
Property: ke
Type: Array

6.6.1.33.118 Key Exchange Method (KE)

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/ke/[]
Type: Object

Object representing a key exchange method (KE).

Table 235 — Properties for the ke object

Property	Type	Requirement	Description
group	Integer	Optional	A group identifier for the key exchange algorithm.
algorithm	String	Optional	The bom-ref to algorithm cryptographic asset.

6.6.1.33.119 Group Identifier

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/ke/[]/group
Type: Integer

A group identifier for the key exchange algorithm.

© Ecma International 2025

6.6.1.33.120 Algorithm reference

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/ke/[]/algorithm
Type: String

The bom-ref to algorithm cryptographic asset.

LEGACY

6.6.1.33.121 Encryption Algorithm (ENCR) Reference Array

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/ke
Property: ke
Type: Array (of String)

[DEPRECATED] This will be removed in a future version. 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 shall be a string.

6.6.1.33.122 Extended Sequence Number (ESN)

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/esn
Type: Boolean

Specifies if an Extended Sequence Number (ESN) is used.

6.6.1.33.123 IKEv2 Authentication methods

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/auth

IKEv2 Authentication method per RFC9593.

Shall be any of:

1. IKEv2 Authentication Methods
2. Encryption Algorithm (ENCR) Reference Array

6.6.1.33.124 IKEv2 Authentication Methods

Type: Array

LEGACY

6.6.1.33.125 Encryption Algorithm (ENCR) Reference Array

Type: Array

[DEPRECATED] This will be removed in a future version. IKEv2 Authentication method.

6.6.1.33.126 IKEv2 Authentication Methods

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/auth
Property: auth
Type: Array

© Ecma International 2025

6.6.1.33.127 IKEv2 Authentication Method

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/auth/[]
Type: Object

Object representing a IKEv2 Authentication method.

Table 236 — Properties for the auth object

Property	Type	Requirement	Description
name	String	Optional	A name for the authentication method.
algorithm	String	Optional	The bom-ref to algorithm cryptographic asset.

6.6.1.33.128 Name

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/auth/[]/name
Type: String

A name for the authentication method.

6.6.1.33.129 Algorithm reference

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/auth/[]/algorithm
Type: String

The bom-ref to algorithm cryptographic asset.

LEGACY

6.6.1.33.130 Encryption Algorithm (ENCR) Reference Array

Location: /components/[]/cryptoProperties/protocolProperties/ikev2TransformTypes/auth
Property: auth
Type: Array (of String)

[DEPRECATED] This will be removed in a future version. 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 shall be a string.

6.6.1.33.131 Cryptographic References

Location: /components/[]/cryptoProperties/protocolProperties/cryptoRefArray
Property: cryptoRefArray (Optional)
Type: Array (of String)

[DEPRECATED] Use **`@.relatedCryptographicAssets`** instead. 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 shall be a string.

© Ecma International 2025

6.6.1.33.132 Related Cryptographic Assets

Location: /components/[]/cryptoProperties/protocolProperties/relatedCryptographicAssets
Property: relatedCryptographicAssets (Optional)
Type: Array

A list of cryptographic assets related to this component. Each item of this array shall be a Related Cryptographic Asset object.

6.6.1.33.133 Related Cryptographic Asset

Location: /components/[]/cryptoProperties/protocolProperties/relatedCryptographicAssets/[]
Type: Object

A cryptographic assets related to this component.

Table 237 — Properties for the relatedCryptographicAsset object

Property	Type	Requirement	Description
type	String	Optional	Specifies the mechanism by which the cryptographic asset is secured by.
ref	String	Optional	The bom-ref to cryptographic asset.

6.6.1.33.134 Type

Location: /components/[]/cryptoProperties/protocolProperties/relatedCryptographicAssets/[]/type
Type: String

Specifies the mechanism by which the cryptographic asset is secured by.

Example 1 (Informative)
```
publicKey
```

Example 2 (Informative)
```
privateKey
```

Example 3 (Informative)
```
algorithm
```

6.6.1.33.135 Reference to cryptographic asset

Location: /components/[]/cryptoProperties/protocolProperties/relatedCryptographicAssets/[]/ref
Type: String

The bom-ref to cryptographic asset.

6.6.1.33.136 OID

Location: /components/[]/cryptoProperties/oid
Type: String

The object identifier (OID) of the cryptographic asset.

© Ecma International 2025

6.6.1.34 Properties

Location: /components/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.6.1.34.1 Lightweight name-value pair

Location: /components/[]/properties/[]
Type: Object

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 238 — Properties for the property 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.

6.6.1.34.2 Name

Location: /components/[]/properties/[]/name
Type: String

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

6.6.1.34.3 Value

Location: /components/[]/properties/[]/value
Type: String

The value of the property.

6.6.1.35 Tags

Location: /components/[]/tags
Property: tags (Optional)
Type: Array (of String)

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 shall be a string.

© Ecma International 2025

Example 1 (Informative)
`json-parser`

Example 2 (Informative)
`object-persistence`

Example 3 (Informative)
`text-to-image`

Example 4 (Informative)
`translation`

Example 5 (Informative)
`object-detection`

6.6.1.36 Signature

Location: /components/[]/signature
Type: Object

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.

6.7 Services

Location: /services
Property: services (Optional)
Type: Array
Uniqueness: All items shall be unique.

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 shall be a Service object.

6.7.1 Service

Location: /services/[]
Type: Object

© Ecma International 2025

Table 239 — Properties for the service object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the service elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. 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.
licenses	Array	Optional	A list of SPDX licenses and/or named licenses and/or SPDX Licence Expression.
patentAssertions	Array	Optional	A list of assertions made regarding patents associated with this component or service. Assertions distinguish between ownership, licensing, and other relevant interactions with patents.
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 release notes.

© Ecma International 2025

Table 239 — Properties for the service object *(continued)*

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.
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).

6.7.1.1 BOM Reference

Location: /services/[]/bom-ref
Type: String

An identifier which can be used to reference the service elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. Value should not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

6.7.1.2 Provider

Location: /services/[]/provider
Type: Object

The organization that provides the service.

Table 240 — Properties for the provider object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.7.1.2.1 BOM Reference

Location: /services/[]/provider/bom-ref
Type: String

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

© Ecma International 2025

6.7.1.2.2 Organization Name

Location: /services/[]/provider/name
Type: String

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.7.1.2.3 Organization Address

Location: /services/[]/provider/address
Type: Object

The physical address (location) of the organization.

Table 241 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.7.1.2.4 BOM Reference

Location: /services/[]/provider/address/bom-ref
Type: String

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

6.7.1.2.5 Country

Location: /services/[]/provider/address/country
Type: String

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

© Ecma International 2025

6.7.1.2.6 Region

Location: /services/[]/provider/address/region
Type: String

The region or state in the country.

Example (Informative)
```
Texas
```

6.7.1.2.7 Locality

Location: /services/[]/provider/address/locality
Type: String

The locality or city within the country.

Example (Informative)
```
Austin
```

6.7.1.2.8 Post Office Box Number

Location: /services/[]/provider/address/postOfficeBoxNumber
Type: String

The post office box number.

Example (Informative)
```
901
```

6.7.1.2.9 Postal Code

Location: /services/[]/provider/address/postalCode
Type: String

The postal code.

Example (Informative)
```
78758
```

6.7.1.2.10 Street Address

Location: /services/[]/provider/address/streetAddress
Type: String

The street address.

Example (Informative)
```
100 Main Street
```


© Ecma International 2025

6.7.1.2.11 Organization URL(s)

Location: /services/[]/provider/url
Property: url (Optional)
Type: Array (of String)
Format: iri-reference as specified in RFC 3987

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

Example (Informative)

```
https://example.com
```

6.7.1.2.12 Organizational Contact

Location: /services/[]/provider/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.7.1.2.13 Organizational Person

Location: /services/[]/provider/contact/[]
Type: Object

Table 242 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.7.1.2.14 BOM Reference

Location: /services/[]/provider/contact/[]/bom-ref
Type: String

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

6.7.1.2.15 Name

Location: /services/[]/provider/contact/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

© Ecma International 2025

6.7.1.2.16 Email Address

Location: /services/[]/provider/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.7.1.2.17 Phone

Location: /services/[]/provider/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.7.1.3 Service Group

Location: /services/[]/group
Type: String

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.

Example (Informative)
```
com.acme
```

6.7.1.4 Service Name

Location: /services/[]/name
Type: String

The name of the service. This will often be a shortened, single name of the service.

Example (Informative)
```
ticker-service
```

6.7.1.5 Service Version

Location: /services/[]/version
Type: String

The service version.

Example 1 (Informative)
```
9.0.14
```


© Ecma International 2025

Example 2 (Informative)
```
v1.33.7
```

Example 3 (Informative)
```
7.0.0-M1
```

Example 4 (Informative)
```
2.0pre1
```

Example 5 (Informative)
```
1.0.0-beta1
```

Example 6 (Informative)
```
0.8.15
```

6.7.1.6 Service Description

Location: /services/[]/description
Type: String

Specifies a description for the service.

6.7.1.7 Endpoints

Location: /services/[]/endpoints
Property: endpoints (Optional)
Type: Array (of String)
Format: iri-reference as specified in RFC 3987

The endpoint URIs of the service. Multiple endpoints are allowed. Each item of this array shall be a string.

Example (Informative)
```
https://example.com/api/v1/ticker
```

6.7.1.8 Authentication Required

Location: /services/[]/authenticated
Type: Boolean

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.

6.7.1.9 Crosses Trust Boundary

Location: /services/[]/x-trust-boundary
Type: Boolean

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.

© Ecma International 2025

6.7.1.10 Trust Zone

Location: /services/[]/trustZone
Type: String

The name of the trust zone the service resides in.

6.7.1.11 Data

Location: /services/[]/data
Property: data (Optional)
Type: Array

Specifies information about the data including the directional flow of data and the data classification. Each item of this array shall be a Hash Objects object.

6.7.1.11.1 Hash Objects

Location: /services/[]/data/[]
Type: Object

Table 243 — 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.
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.

6.7.1.11.2 Directional Flow

Location: /services/[]/data/[]/flow
Type: String (enum)

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.

© Ecma International 2025

Table 244 — 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.

6.7.1.11.3 Data Classification

Location: /services/[]/data/[]/classification
Type: String

Data classification tags data according to its type, sensitivity, and value if altered, stolen, or destroyed.

6.7.1.11.4 Name

Location: /services/[]/data/[]/name
Type: String

Name for the defined data.

Example (Informative)

```
Credit card reporting
```

6.7.1.11.5 Description

Location: /services/[]/data/[]/description
Type: String

Short description of the data content and usage.

Example (Informative)

```
Credit card information being exchanged in between the web app and the
database
```

6.7.1.11.6 Data Governance

Location: /services/[]/data/[]/governance
Type: Object

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.

© Ecma International 2025

Table 245 — 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.

6.7.1.11.7 Data Custodians

Location: /services/[]/data/[]/governance/custodians
Property: custodians (Optional)
Type: Array

Data custodians are responsible for the safe custody, transport, and storage of data.

6.7.1.11.8 Custodian

Location: /services/[]/data/[]/governance/custodians/[]
Type: Object

Table 246 — Properties for the custodian 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).

6.7.1.11.9 Organization

Location: /services/[]/data/[]/governance/custodians/[]/organization
Type: Object

The organization that is responsible for specific data governance role(s).

Table 247 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

© Ecma International 2025

6.7.1.11.10 BOM Reference

Location: /services/[]/data/[]/governance/custodians/[]/organization/bom-ref
Type: String

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

6.7.1.11.11 Organization Name

Location: /services/[]/data/[]/governance/custodians/[]/organization/name
Type: String

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.7.1.11.12 Organization Address

Location: /services/[]/data/[]/governance/custodians/[]/organization/address
Type: Object

The physical address (location) of the organization.

Table 248 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.7.1.11.13 BOM Reference

Location: /services/[]/data/[]/governance/custodians/[]/organization/address/bom-ref
Type: String

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

© Ecma International 2025

6.7.1.11.14 Country

Location: /services/[]/data/[]/governance/custodians/[]/organization/address/country
Type: String

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

6.7.1.11.15 Region

Location: /services/[]/data/[]/governance/custodians/[]/organization/address/region
Type: String

The region or state in the country.

Example (Informative)

```
Texas
```

6.7.1.11.16 Locality

Location: /services/[]/data/[]/governance/custodians/[]/organization/address/locality
Type: String

The locality or city within the country.

Example (Informative)

```
Austin
```

6.7.1.11.17 Post Office Box Number

Location: /services/[]/data/[]/governance/custodians/[]/organization/address/postOfficeBoxNumber
Type: String

The post office box number.

Example (Informative)

```
901
```

6.7.1.11.18 Postal Code

Location: /services/[]/data/[]/governance/custodians/[]/organization/address/postalCode
Type: String

The postal code.

Example (Informative)

```
78758
```


© Ecma International 2025

6.7.1.11.19 Street Address

Location: /services/[]/data/[]/governance/custodians/[]/organization/address/streetAddress
Type: String

The street address.

Example (Informative)
```
100 Main Street
```

6.7.1.11.20 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

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

Example (Informative)
```
https://example.com
```

6.7.1.11.21 Organizational Contact

Location: /services/[]/data/[]/governance/custodians/[]/organization/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.7.1.11.22 Organizational Person

Location: /services/[]/data/[]/governance/custodians/[]/organization/contact/[]
Type: Object

Table 249 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.7.1.11.23 BOM Reference

Location: /services/[]/data/[]/governance/custodians/[]/organization/contact/[]/bom-ref
Type: String

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

© Ecma International 2025

6.7.1.11.24 Name

Location: /services/[]/data/[]/governance/custodians/[]/organization/contact/[]/name
Type: String

The name of a contact.

Example (Informative)
```
Contact name
```

6.7.1.11.25 Email Address

Location: /services/[]/data/[]/governance/custodians/[]/organization/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.7.1.11.26 Phone

Location: /services/[]/data/[]/governance/custodians/[]/organization/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.7.1.11.27 Individual

Location: /services/[]/data/[]/governance/custodians/[]/contact
Type: Object

The individual that is responsible for specific data governance role(s).

Table 250 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

© Ecma International 2025

6.7.1.11.28 BOM Reference

Location: /services/[]/data/[]/governance/custodians/[]/contact/bom-ref
Type: String

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

6.7.1.11.29 Name

Location: /services/[]/data/[]/governance/custodians/[]/contact/name
Type: String

The name of a contact.

Example (Informative)
```
Contact name
```

6.7.1.11.30 Email Address

Location: /services/[]/data/[]/governance/custodians/[]/contact/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.7.1.11.31 Phone

Location: /services/[]/data/[]/governance/custodians/[]/contact/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.7.1.11.32 Data Stewards

Location: /services/[]/data/[]/governance/stewards
Property: stewards (Optional)
Type: Array

Data stewards are responsible for data content, context, and associated business rules.

6.7.1.11.33 Steward

Location: /services/[]/data/[]/governance/stewards/[]
Type: Object

© Ecma International 2025

Table 251 — Properties for the steward 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).

6.7.1.11.34 Organization

Location: /services/[]/data/[]/governance/stewards/[]/organization
Type: Object

The organization that is responsible for specific data governance role(s).

Table 252 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.7.1.11.35 BOM Reference

Location: /services/[]/data/[]/governance/stewards/[]/organization/bom-ref
Type: String

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

6.7.1.11.36 Organization Name

Location: /services/[]/data/[]/governance/stewards/[]/organization/name
Type: String

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.7.1.11.37 Organization Address

Location: /services/[]/data/[]/governance/stewards/[]/organization/address
Type: Object

The physical address (location) of the organization.

© Ecma International 2025

Table 253 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.7.1.11.38 BOM Reference

Location: /services/[]/data/[]/governance/stewards/[]/organization/address/bom-ref
Type: String

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

6.7.1.11.39 Country

Location: /services/[]/data/[]/governance/stewards/[]/organization/address/country
Type: String

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

6.7.1.11.40 Region

Location: /services/[]/data/[]/governance/stewards/[]/organization/address/region
Type: String

The region or state in the country.

Example (Informative)

```
Texas
```

6.7.1.11.41 Locality

Location: /services/[]/data/[]/governance/stewards/[]/organization/address/locality
Type: String

The locality or city within the country.

Example (Informative)

```
Austin
```

© Ecma International 2025

6.7.1.11.42 Post Office Box Number

Location: /services/[]/data/[]/governance/stewards/[]/organization/address/postOfficeBoxNumber
Type: String

The post office box number.

Example (Informative)
```
901
```

6.7.1.11.43 Postal Code

Location: /services/[]/data/[]/governance/stewards/[]/organization/address/postalCode
Type: String

The postal code.

Example (Informative)
```
78758
```

6.7.1.11.44 Street Address

Location: /services/[]/data/[]/governance/stewards/[]/organization/address/streetAddress
Type: String

The street address.

Example (Informative)
```
100 Main Street
```

6.7.1.11.45 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

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

Example (Informative)
```
https://example.com
```

6.7.1.11.46 Organizational Contact

Location: /services/[]/data/[]/governance/stewards/[]/organization/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.7.1.11.47 Organizational Person

Location: /services/[]/data/[]/governance/stewards/[]/organization/contact/[]
Type: Object

© Ecma International 2025

Table 254 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.7.1.11.48 BOM Reference

Location: /services/[]/data/[]/governance/stewards/[]/organization/contact/[]/bom-ref
Type: String

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

6.7.1.11.49 Name

Location: /services/[]/data/[]/governance/stewards/[]/organization/contact/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.7.1.11.50 Email Address

Location: /services/[]/data/[]/governance/stewards/[]/organization/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

6.7.1.11.51 Phone

Location: /services/[]/data/[]/governance/stewards/[]/organization/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)

```
800-555-1212
```

© Ecma International 2025

6.7.1.11.52 Individual

Location: /services/[]/data/[]/governance/stewards/[]/contact
Type: Object

The individual that is responsible for specific data governance role(s).

Table 255 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.7.1.11.53 BOM Reference

Location: /services/[]/data/[]/governance/stewards/[]/contact/bom-ref
Type: String

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

6.7.1.11.54 Name

Location: /services/[]/data/[]/governance/stewards/[]/contact/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.7.1.11.55 Email Address

Location: /services/[]/data/[]/governance/stewards/[]/contact/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```


© Ecma International 2025

6.7.1.11.56 Phone

Location: /services/[]/data/[]/governance/stewards/[]/contact/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.7.1.11.57 Data Owners

Location: /services/[]/data/[]/governance/owners
Property: owners (Optional)
Type: Array

Data owners are concerned with risk and appropriate access to data.

6.7.1.11.58 Owner

Location: /services/[]/data/[]/governance/owners/[]
Type: Object

Table 256 — Properties for the owner 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).

6.7.1.11.59 Organization

Location: /services/[]/data/[]/governance/owners/[]/organization
Type: Object

The organization that is responsible for specific data governance role(s).

Table 257 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

© Ecma International 2025

6.7.1.11.60 BOM Reference

Location: /services/[]/data/[]/governance/owners/[]/organization/bom-ref
Type: String

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

6.7.1.11.61 Organization Name

Location: /services/[]/data/[]/governance/owners/[]/organization/name
Type: String

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.7.1.11.62 Organization Address

Location: /services/[]/data/[]/governance/owners/[]/organization/address
Type: Object

The physical address (location) of the organization.

Table 258 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.7.1.11.63 BOM Reference

Location: /services/[]/data/[]/governance/owners/[]/organization/address/bom-ref
Type: String

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

© Ecma International 2025

6.7.1.11.64 Country

Location: /services/[]/data/[]/governance/owners/[]/organization/address/country
Type: String

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

6.7.1.11.65 Region

Location: /services/[]/data/[]/governance/owners/[]/organization/address/region
Type: String

The region or state in the country.

Example (Informative)

```
Texas
```

6.7.1.11.66 Locality

Location: /services/[]/data/[]/governance/owners/[]/organization/address/locality
Type: String

The locality or city within the country.

Example (Informative)

```
Austin
```

6.7.1.11.67 Post Office Box Number

Location: /services/[]/data/[]/governance/owners/[]/organization/address/postOfficeBoxNumber
Type: String

The post office box number.

Example (Informative)

```
901
```

6.7.1.11.68 Postal Code

Location: /services/[]/data/[]/governance/owners/[]/organization/address/postalCode
Type: String

The postal code.

Example (Informative)

```
78758
```

© Ecma International 2025

6.7.1.11.69 Street Address

Location: /services/[]/data/[]/governance/owners/[]/organization/address/streetAddress
Type: String

The street address.

Example (Informative)

```
100 Main Street
```

6.7.1.11.70 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

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

Example (Informative)

```
https://example.com
```

6.7.1.11.71 Organizational Contact

Location: /services/[]/data/[]/governance/owners/[]/organization/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.7.1.11.72 Organizational Person

Location: /services/[]/data/[]/governance/owners/[]/organization/contact/[]
Type: Object

Table 259 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.7.1.11.73 BOM Reference

Location: /services/[]/data/[]/governance/owners/[]/organization/contact/[]/bom-ref
Type: String

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

© Ecma International 2025

6.7.1.11.74 Name

Location: /services/[]/data/[]/governance/owners/[]/organization/contact/[]/name
Type: String

The name of a contact.

Example (Informative)
```
Contact name
```

6.7.1.11.75 Email Address

Location: /services/[]/data/[]/governance/owners/[]/organization/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.7.1.11.76 Phone

Location: /services/[]/data/[]/governance/owners/[]/organization/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.7.1.11.77 Individual

Location: /services/[]/data/[]/governance/owners/[]/contact
Type: Object

The individual that is responsible for specific data governance role(s).

Table 260 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

© Ecma International 2025

6.7.1.11.78 BOM Reference

Location: /services/[]/data/[]/governance/owners/[]/contact/bom-ref
Type: String

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

6.7.1.11.79 Name

Location: /services/[]/data/[]/governance/owners/[]/contact/name
Type: String

The name of a contact.

Example (Informative)
```
Contact name
```

6.7.1.11.80 Email Address

Location: /services/[]/data/[]/governance/owners/[]/contact/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.7.1.11.81 Phone

Location: /services/[]/data/[]/governance/owners/[]/contact/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.7.1.11.82 Source

Location: /services/[]/data/[]/source
Property: source (Optional)
Type: Array

The URI, URL, or BOM-Link of the components or services the data came in from

6.7.1.11.83 Source

Location: /services/[]/data/[]/source/[]

Shall be any of:

1. URL
2. BOM-Link Element

© Ecma International 2025

6.7.1.11.84 URL

Type: String
Format: iri-reference as specified in RFC 3987

6.7.1.11.85 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.7.1.11.86 Destination

Location: /services/[]/data/[]/destination
Property: destination (Optional)
Type: Array

The URI, URL, or BOM-Link of the components or services the data is sent to

6.7.1.11.87 Destination

Location: /services/[]/data/[]/destination/[]

Shall be any of:

1. URL
2. BOM-Link Element

6.7.1.11.88 URL

Type: String
Format: iri-reference as specified in RFC 3987

6.7.1.11.89 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.7.1.12 Service License(s)

Location: /services/[]/licenses
Property: licenses (Optional)
Type: Array

A list of SPDX licenses and/or named licenses and/or SPDX Licence Expression.

6.7.1.12.1 License

Location: /services/[]/licenses/[]

© Ecma International 2025

6.7.1.13 Service Patent(s)

Location: /services/[]/patentAssertions
Property: patentAssertions (Optional)
Type: Array

A list of assertions made regarding patents associated with this component or service. Assertions distinguish between ownership, licensing, and other relevant interactions with patents. Each item of this array shall be a Patent Assertion object.

6.7.1.13.1 Patent Assertion

Location: /services/[]/patentAssertions/[]
Type: Object

An assertion linking a patent or patent family to this component or service.

Table 261 — Properties for the patentAssertion object

Property	Type	Requirement	Description
bom-ref	String	Optional	A reference to the patent or patent family object within the BOM. This shall match the **`bom-ref`** of a **`patent`** or **`patentFamily`** object.
assertionType	String	Required	The type of assertion being made about the patent or patent family. Examples include ownership, licensing, and standards inclusion.
patentRefs	Array	Optional	A list of BOM references (**`bom-ref`**) linking to patents or patent families associated with this assertion.
asserter	Array	Required	undefined.
notes	String	Optional	Additional notes or clarifications regarding the assertion, if necessary. For example, geographical restrictions, duration, or limitations of a licence.

6.7.1.13.2 BOM Reference

Location: /services/[]/patentAssertions/[]/bom-ref
Type: String

A reference to the patent or patent family object within the BOM. This shall match the **`bom-ref`** of a **`patent`** or **`patentFamily`** object.

6.7.1.13.3 Assertion Type

Location: /services/[]/patentAssertions/[]/assertionType
Type: String (enum)

The type of assertion being made about the patent or patent family. Examples include ownership, licensing, and standards inclusion.

© Ecma International 2025

Table 262 — Enumeration of possible values

Value	Description
ownership	The manufacturer asserts ownership of the patent or patent family.
license	The manufacturer asserts they have a licence to use the patent or patent family.
third-party-claim	A third party has asserted a claim or potential infringement against the manufacturer's component or service.
standards-inclusion	The patent is part of a standard essential patent (SEP) portfolio relevant to the component or service.
prior-art	The manufacturer asserts the patent or patent family as prior art that invalidates another patent or claim.
exclusive-rights	The manufacturer asserts exclusive rights granted through a licensing agreement.
non-assertion	The manufacturer asserts they will not enforce the patent or patent family against certain uses or users.
research-or-evaluation	The patent or patent family is being used under a research or evaluation licence.

6.7.1.13.4 Patent References

Location: /services/[]/patentAssertions/[]/patentRefs
Property: patentRefs (Optional)
Type: Array (of String)

A list of BOM references (**`bom-ref`**) linking to patents or patent families associated with this assertion. 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 shall be a string.

6.7.1.13.5 Asserter

Location: /services/[]/patentAssertions/[]/asserter

Shall be one of:

1. Organizational Entity
2. Person
3. Reference

6.7.1.13.6 Organizational Entity

Type: Object

Table 263 — Properties for the organizational entity object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

© Ecma International 2025

Table 263 — Properties for the organizational entity object *(continued)*

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.

6.7.1.13.7 Person

Type: Object

Table 264 — Properties for the person object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.7.1.13.8 Reference

Type: String

A reference to a previously defined **`organizationalContact`** or **`organizationalEntity`** object in the BOM. The value shall be a valid **`bom-ref`** pointing to one of these objects.

6.7.1.13.9 BOM Reference

Location: /services/[]/patentAssertions/[]/asserter/bom-ref
Property: bom-ref (Optional)
Type: String

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

6.7.1.13.10 Organization Name

Location: /services/[]/patentAssertions/[]/asserter/name
Property: name (Optional)
Type: String

The name of the organization.

Example (Informative)

```
Example Inc.
```


© Ecma International 2025

6.7.1.13.11 Organization Address

Location: /services/[]/patentAssertions/[]/asserter/address
Property: address (Optional)
Type: Object

The physical address (location) of the organization.

Table 265 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.7.1.13.12 BOM Reference

Location: /services/[]/patentAssertions/[]/asserter/address/bom-ref
Property: bom-ref (Optional)
Type: String

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

6.7.1.13.13 Country

Location: /services/[]/patentAssertions/[]/asserter/address/country
Property: country (Optional)
Type: String

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

6.7.1.13.14 Region

Location: /services/[]/patentAssertions/[]/asserter/address/region
Property: region (Optional)
Type: String

The region or state in the country.

Example (Informative)

```
Texas
```

© Ecma International 2025

6.7.1.13.15 Locality

Location: /services/[]/patentAssertions/[]/asserter/address/locality
Property: locality (Optional)
Type: String

The locality or city within the country.

Example (Informative)
```
Austin
```

6.7.1.13.16 Post Office Box Number

Location: /services/[]/patentAssertions/[]/asserter/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)
Type: String

The post office box number.

Example (Informative)
```
901
```

6.7.1.13.17 Postal Code

Location: /services/[]/patentAssertions/[]/asserter/address/postalCode
Property: postalCode (Optional)
Type: String

The postal code.

Example (Informative)
```
78758
```

6.7.1.13.18 Street Address

Location: /services/[]/patentAssertions/[]/asserter/address/streetAddress
Property: streetAddress (Optional)
Type: String

The street address.

Example (Informative)
```
100 Main Street
```

6.7.1.13.19 Organization URL(s)

Location: /services/[]/patentAssertions/[]/asserter/url
Property: url (Optional)
Type: Array (of String)
Format: iri-reference as specified in RFC 3987

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

Example (Informative)
```
https://example.com
```


© Ecma International 2025

6.7.1.13.20 Organizational Contact

Location: /services/[]/patentAssertions/[]/asserter/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.7.1.13.21 Organizational Person

Location: /services/[]/patentAssertions/[]/asserter/contact/[]
Type: Object

Table 266 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.7.1.13.22 BOM Reference

Location: /services/[]/patentAssertions/[]/asserter/contact/[]/bom-ref
Type: String

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

6.7.1.13.23 Name

Location: /services/[]/patentAssertions/[]/asserter/contact/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.7.1.13.24 Email Address

Location: /services/[]/patentAssertions/[]/asserter/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

© Ecma International 2025

6.7.1.13.25 Phone

Location: /services/[]/patentAssertions/[]/asserter/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.7.1.13.26 BOM Reference

Location: /services/[]/patentAssertions/[]/asserter/bom-ref
Property: bom-ref (Optional)
Type: String

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

6.7.1.13.27 Name

Location: /services/[]/patentAssertions/[]/asserter/name
Property: name (Optional)
Type: String

The name of a contact.

Example (Informative)
```
Contact name
```

6.7.1.13.28 Email Address

Location: /services/[]/patentAssertions/[]/asserter/email
Property: email (Optional)
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.7.1.13.29 Phone

Location: /services/[]/patentAssertions/[]/asserter/phone
Property: phone (Optional)
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```


© Ecma International 2025

6.7.1.13.30 Notes

Location: /services/[]/patentAssertions/[]/notes
Type: String

Additional notes or clarifications regarding the assertion, if necessary. For example, geographical restrictions, duration, or limitations of a licence.

6.7.1.14 External References

Location: /services/[]/externalReferences
Property: externalReferences (Optional)
Type: Array

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 shall be an External Reference object.

6.7.1.14.1 External Reference

Location: /services/[]/externalReferences/[]
Type: Object

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.

6.7.1.15 Services

Location: /services/[]/services
Property: services (Optional)
Type: Array
Uniqueness: All items shall be unique.

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 shall be a Service object.

6.7.1.15.1 Service

Location: /services/[]/services/[]
Type: Object

6.7.1.16 Release notes

Location: /services/[]/releaseNotes
Type: Object

Specifies release notes.

© Ecma International 2025

Table 267 — 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.

6.7.1.16.1 Type

Location: /services/[]/releaseNotes/type
Type: String

The software versioning type the release note describes.

Example 1 (Informative)

```
major
```

Example 2 (Informative)

```
minor
```

Example 3 (Informative)

```
patch
```

Example 4 (Informative)

```
pre-release
```


© Ecma International 2025

Example 5 (Informative)
`internal`

6.7.1.16.2 Title

Location: /services/[]/releaseNotes/title
Type: String

The title of the release.

6.7.1.16.3 Featured image

Location: /services/[]/releaseNotes/featuredImage
Type: String
Format: iri-reference as specified in RFC 3987

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

6.7.1.16.4 Social image

Location: /services/[]/releaseNotes/socialImage
Type: String
Format: iri-reference as specified in RFC 3987

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

6.7.1.16.5 Description

Location: /services/[]/releaseNotes/description
Type: String

A short description of the release.

6.7.1.16.6 Timestamp

Location: /services/[]/releaseNotes/timestamp
Type: String
Format: data-time as specified in RFC 3339 section 5.6

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

6.7.1.16.7 Aliases

Location: /services/[]/releaseNotes/aliases
Property: aliases (Optional)
Type: Array (of String)

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 shall be a string.

© Ecma International 2025

6.7.1.16.8 Tags

Location: /services/[]/releaseNotes/tags
Property: tags (Optional)
Type: Array (of String)

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 shall be a string.

Example 1 (Informative)
```
json-parser
```

Example 2 (Informative)
```
object-persistence
```

Example 3 (Informative)
```
text-to-image
```

Example 4 (Informative)
```
translation
```

Example 5 (Informative)
```
object-detection
```

6.7.1.16.9 Resolves

Location: /services/[]/releaseNotes/resolves
Property: resolves (Optional)
Type: Array

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

6.7.1.16.10 Issue

Location: /services/[]/releaseNotes/resolves/[]
Type: Object

An individual issue that has been resolved.

Table 268 — Properties for the resolve 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.

© Ecma International 2025

6.7.1.16.11 Issue Type

Location: /services/[]/releaseNotes/resolves/[]/type
Type: String (enum)

Specifies the type of issue.

Table 269 — 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.

6.7.1.16.12 Issue ID

Location: /services/[]/releaseNotes/resolves/[]/id
Type: String

The identifier of the issue assigned by the source of the issue.

6.7.1.16.13 Issue Name

Location: /services/[]/releaseNotes/resolves/[]/name
Type: String

The name of the issue.

6.7.1.16.14 Issue Description

Location: /services/[]/releaseNotes/resolves/[]/description
Type: String

A description of the issue.

6.7.1.16.15 Source

Location: /services/[]/releaseNotes/resolves/[]/source
Type: Object

The source of the issue where it is documented.

Table 270 — 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.

© Ecma International 2025

6.7.1.16.16 Name

Location: /services/[]/releaseNotes/resolves/[]/source/name
Type: String

The name of the source.

Example 1 (Informative)
```
National Vulnerability Database
```

Example 2 (Informative)
```
NVD
```

Example 3 (Informative)
```
Apache
```

6.7.1.16.17 URL

Location: /services/[]/releaseNotes/resolves/[]/source/url
Type: String
Format: iri-reference as specified in RFC 3987

The url of the issue documentation as provided by the source.

6.7.1.16.18 References

Location: /services/[]/releaseNotes/resolves/[]/references
Property: references (Optional)
Type: Array (of String)
Format: iri-reference as specified in RFC 3987

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

Example (Informative)
```
https://example.com
```

6.7.1.16.19 Notes

Location: /services/[]/releaseNotes/notes
Property: notes (Optional)
Type: Array

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 shall be a Note object.

6.7.1.16.20 Note

Location: /services/[]/releaseNotes/notes/[]
Type: Object

A note containing the locale and content.

© Ecma International 2025

Table 271 — Properties for the note 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.

6.7.1.16.21 Locale

Location: /services/[]/releaseNotes/notes/[]/locale
Type: String
Pattern Constraint: `^([a-z]{2})(-[A-Z]{2})?$`

The ISO-639 (or higher) language code and optional ISO-3166 (or higher) country code. Examples include: "en", "en-US", "fr" and "fr-CA".

6.7.1.16.22 Release note content

Location: /services/[]/releaseNotes/notes/[]/text
Type: Object

Specifies the full content of the release note.

Table 272 — 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 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.

6.7.1.16.23 Content-Type

Location: /services/[]/releaseNotes/notes/[]/text/contentType
Type: String
Default Value: text/plain

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.

Example 1 (Informative)

```
text/plain
```

Example 2 (Informative)

```
application/json
```

© Ecma International 2025

Example 3 (Informative)
`image/png`

6.7.1.16.24 Encoding

Location: /services/[]/releaseNotes/notes/[]/text/encoding
Type: String (enum)

Specifies the encoding the text is represented in.

Table 273 — Enumeration of possible values

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

6.7.1.16.25 Attachment Text

Location: /services/[]/releaseNotes/notes/[]/text/content
Type: String

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

6.7.1.16.26 Properties

Location: /services/[]/releaseNotes/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.7.1.16.27 Lightweight name-value pair

Location: /services/[]/releaseNotes/properties/[]
Type: Object

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 274 — Properties for the property 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.

© Ecma International 2025

6.7.1.16.28 Name

Location: /services/[]/releaseNotes/properties/[]/name
Type: String

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

6.7.1.16.29 Value

Location: /services/[]/releaseNotes/properties/[]/value
Type: String

The value of the property.

6.7.1.17 Properties

Location: /services/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.7.1.17.1 Lightweight name-value pair

Location: /services/[]/properties/[]
Type: Object

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 property 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.

6.7.1.17.2 Name

Location: /services/[]/properties/[]/name
Type: String

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

© Ecma International 2025

6.7.1.17.3 Value

Location: /services/[]/properties/[]/value
Type: String

The value of the property.

6.7.1.18 Tags

Location: /services/[]/tags
Property: tags (Optional)
Type: Array (of String)

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 shall be a string.

Example 1 (Informative)
```
json-parser
```

Example 2 (Informative)
```
object-persistence
```

Example 3 (Informative)
```
text-to-image
```

Example 4 (Informative)
```
translation
```

Example 5 (Informative)
```
object-detection
```

6.7.1.19 Signature

Location: /services/[]/signature
Type: Object

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.

6.8 External References

Location: /externalReferences
Property: externalReferences (Optional)
Type: Array

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 shall be an External Reference object.

© Ecma International 2025

6.8.1 External Reference

Location: /externalReferences/[]
Type: Object

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 276 — Properties for the externalReference 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	A 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).
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.

6.8.1.1 URL

Location: /externalReferences/[]/url

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.

Shall be any of:

1. URL
2. BOM-Link

6.8.1.2 URL

Type: String
Format: iri-reference as specified in RFC 3987

6.8.1.3 BOM-Link

Type: Object

© Ecma International 2025

6.8.1.4 Comment

Location: /externalReferences/[]/comment
Type: String

A comment describing the external reference.

6.8.1.5 Type

Location: /externalReferences/[]/type
Type: String (enum)

Specifies the type of external reference.

Table 277 — 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.

© Ecma International 2025

Table 277 — Enumeration of possible values *(continued)*

Value	Description
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.
evidence	Information used to substantiate a claim.
formulation	Describes the formulation of any referencable object within the BOM, including components, services, metadata, declarations, or the BOM itself.
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 analyzes 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).

© Ecma International 2025

Table 277 — Enumeration of possible values *(continued)*

Value	Description
quality-metrics	Report or system in which quality metrics can be obtained.
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).
patent	References information about patents which may be defined in human-readable documents or in machine-readable formats such as CycloneDX or ST.96. For detailed patent information or to reference the information provided directly by patent offices, it is recommended to leverage standards from the World Intellectual Property Organization (WIPO) such as ST.96.
patent-family	References information about a patent family which may be defined in human-readable documents or in machine-readable formats such as CycloneDX or ST.96. A patent family is a group of related patent applications or granted patents that cover the same or similar invention. For detailed patent family information or to reference the information provided directly by patent offices, it is recommended to leverage standards from the World Intellectual Property Organization (WIPO) such as ST.96.
patent-assertion	References assertions made regarding patents associated with a component or service. Assertions distinguish between ownership, licensing, and other relevant interactions with patents.
citation	A reference to external citations applicable to the object identified by this BOM entry or the BOM itself. When used with a BOM-Link, this allows offloading citations into a separate CycloneDX BOM.
other	Use this if no other types accurately describe the purpose of the external reference.

6.8.1.6 Hashes

Location: /externalReferences/[]/hashes
Property: hashes (Optional)
Type: Array

The hashes of the external reference (if applicable). Each item of this array shall be a Hash object.

6.8.1.6.1 Hash

Location: /externalReferences/[]/hashes/[]
Type: Object

© Ecma International 2025

Table 278 — Properties for the hash object

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

6.8.1.6.2 Hash Algorithm

Location: /externalReferences/[]/hashes/[]/alg
Type: String (enum)

The algorithm that generated the hash value.

Enumeration of possible values:

- MD5
- SHA-1
- SHA-256
- SHA-384
- SHA-512
- SHA3-256
- SHA3-384
- SHA3-512
- BLAKE2b-256
- BLAKE2b-384
- BLAKE2b-512
- BLAKE3
- Streebog-256
- Streebog-512

6.8.1.6.3 Hash Value

Location: /externalReferences/[]/hashes/[]/content
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})$`

The value of the hash.

Example (Informative)
`3942447fac867ae5cdb3229b658f4d48`

6.8.1.7 Properties

Location: /externalReferences/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

© Ecma International 2025

6.8.1.7.1 Lightweight name-value pair

Location: /externalReferences/[]/properties/[]
Type: Object

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 279 — Properties for the property 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.

6.8.1.7.2 Name

Location: /externalReferences/[]/properties/[]/name
Type: String

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

6.8.1.7.3 Value

Location: /externalReferences/[]/properties/[]/value
Type: String

The value of the property.

6.9 Dependencies

Location: /dependencies
Property: dependencies (Optional)
Type: Array
Uniqueness: All items shall be unique.

Provides the ability to document dependency relationships including provided & implemented components. Each item of this array shall be a Dependency object.

6.9.1 Dependency

Location: /dependencies/[]
Type: Object

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 shall 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.

© Ecma International 2025

Table 280 — Properties for the dependency 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.

6.9.1.1 Reference

Location: /dependencies/[]/ref
Type: String

References a component or service by its bom-ref attribute.

6.9.1.2 Depends On

Location: /dependencies/[]/dependsOn
Property: dependsOn (Optional)
Type: Array (of String)
Uniqueness: All items shall be unique.

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 shall be a string.

6.9.1.3 Provides

Location: /dependencies/[]/provides
Property: provides (Optional)
Type: Array (of String)
Uniqueness: All items shall be unique.

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 shall be a string.

6.10 Compositions

Location: /compositions
Property: compositions (Optional)
Type: Array
Uniqueness: All items shall be unique.

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 shall be a Compositions object.

© Ecma International 2025

6.10.1 Compositions

Location: /compositions/[]
Type: Object

Table 281 — Properties for the composition object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the composition elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. 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 describes 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).

6.10.1.1 BOM Reference

Location: /compositions/[]/bom-ref
Type: String

An identifier which can be used to reference the composition elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. Value should not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

6.10.1.2 Aggregate

Location: /compositions/[]/aggregate
Type: String (enum)
Default Value: not_specified

Specifies an aggregate type that describes how complete a relationship is.

© Ecma International 2025

Table 282 — 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.
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.

6.10.1.3 BOM references

Location: /compositions/[]/assemblies
Property: assemblies (Optional)
Type: Array
Uniqueness: All items shall be unique.

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.

6.10.1.3.1 Assembly

Location: /compositions/[]/assemblies/[]

Shall be any of:

1. Ref
2. BOM-Link Element

© Ecma International 2025

6.10.1.3.2 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.10.1.3.3 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.10.1.4 BOM references

Location: /compositions/[]/dependencies
Property: dependencies (Optional)
Type: Array (of String)
Uniqueness: All items shall be unique.

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 shall be a string.

6.10.1.5 BOM references

Location: /compositions/[]/vulnerabilities
Property: vulnerabilities (Optional)
Type: Array (of String)
Uniqueness: All items shall be unique.

The bom-ref identifiers of the vulnerabilities being described. Each item of this array shall be a string.

6.10.1.6 Signature

Location: /compositions/[]/signature
Type: Object

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.

6.11 Vulnerabilities

Location: /vulnerabilities
Property: vulnerabilities (Optional)
Type: Array
Uniqueness: All items shall be unique.

Vulnerabilities identified in components or services. Each item of this array shall be a Vulnerability object.

© Ecma International 2025

6.11.1 Vulnerability

Location: /vulnerabilities/[]
Type: Object

Defines a weakness in a component or service that could be exploited or triggered by a threat source.

Table 283 — Properties for the vulnerability object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the vulnerability elsewhere in the BOM. Every **`bom-ref`** shall 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	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.

© Ecma International 2025

Table 283 — Properties for the vulnerability object *(continued)*

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.

6.11.1.1 BOM Reference

Location: /vulnerabilities/[]/bom-ref
Type: String

An identifier which can be used to reference the vulnerability elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. Value should not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

6.11.1.2 ID

Location: /vulnerabilities/[]/id
Type: String

The identifier that uniquely identifies the vulnerability.

Example 1 (Informative)
```
CVE-2021-39182
```

Example 2 (Informative)
```
GHSA-35m5-8cvj-8783
```

Example 3 (Informative)
```
SNYK-PYTHON-ENROCRYPT-1912876
```

6.11.1.3 Source

Location: /vulnerabilities/[]/source
Type: Object

The source that published the vulnerability.

Table 284 — 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.

© Ecma International 2025

6.11.1.3.1 URL

Location: /vulnerabilities/[]/source/url
Type: String

The url of the vulnerability documentation as provided by the source.

Example (Informative)
```
https://nvd.nist.gov/vuln/detail/CVE-2021-39182
```

6.11.1.3.2 Name

Location: /vulnerabilities/[]/source/name
Type: String

The name of the source.

Example 1 (Informative)
```
NVD
```

Example 2 (Informative)
```
National Vulnerability Database
```

Example 3 (Informative)
```
OSS Index
```

Example 4 (Informative)
```
VulnDB
```

Example 5 (Informative)
```
GitHub Advisories
```

6.11.1.4 References

Location: /vulnerabilities/[]/references
Property: references (Optional)
Type: Array

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.

6.11.1.4.1 Reference

Location: /vulnerabilities/[]/references/[]
Type: Object

© Ecma International 2025

Table 285 — Properties for the reference object

Property	Type	Requirement	Description
id	String	Required	An identifier that uniquely identifies the vulnerability.
source	Object	Required	The source that published the vulnerability.

6.11.1.4.2 ID

Location: /vulnerabilities/[]/references/[]/id
Type: String

An identifier that uniquely identifies the vulnerability.

Example 1 (Informative)
```
CVE-2021-39182
```

Example 2 (Informative)
```
GHSA-35m5-8cvj-8783
```

Example 3 (Informative)
```
SNYK-PYTHON-ENROCRYPT-1912876
```

6.11.1.4.3 Source

Location: /vulnerabilities/[]/references/[]/source
Type: Object

The source that published the vulnerability.

Table 286 — 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.

6.11.1.4.4 URL

Location: /vulnerabilities/[]/references/[]/source/url
Type: String

The url of the vulnerability documentation as provided by the source.

Example (Informative)
```
https://nvd.nist.gov/vuln/detail/CVE-2021-39182
```

6.11.1.4.5 Name

Location: /vulnerabilities/[]/references/[]/source/name
Type: String

The name of the source.

© Ecma International 2025

Example 1 (Informative)
```
NVD
```

Example 2 (Informative)
```
National Vulnerability Database
```

Example 3 (Informative)
```
OSS Index
```

Example 4 (Informative)
```
VulnDB
```

Example 5 (Informative)
```
GitHub Advisories
```

6.11.1.5 Ratings

Location: /vulnerabilities/[]/ratings
Property: ratings (Optional)
Type: Array

List of vulnerability ratings Each item of this array shall be a Rating object.

6.11.1.5.1 Rating

Location: /vulnerabilities/[]/ratings/[]
Type: Object

Defines the severity or risk ratings of a vulnerability.

Table 287 — Properties for the rating 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	A reason for rating the vulnerability as it was.

6.11.1.5.2 Source

Location: /vulnerabilities/[]/ratings/[]/source
Type: Object

The source that calculated the severity or risk rating of the vulnerability.

© Ecma International 2025

Table 288 — 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.

6.11.1.5.3 URL

Location: /vulnerabilities/[]/ratings/[]/source/url
Type: String

The url of the vulnerability documentation as provided by the source.

Example (Informative)
```
https://nvd.nist.gov/vuln/detail/CVE-2021-39182
```

6.11.1.5.4 Name

Location: /vulnerabilities/[]/ratings/[]/source/name
Type: String

The name of the source.

Example 1 (Informative)
```
NVD
```

Example 2 (Informative)
```
National Vulnerability Database
```

Example 3 (Informative)
```
OSS Index
```

Example 4 (Informative)
```
VulnDB
```

Example 5 (Informative)
```
GitHub Advisories
```

6.11.1.5.5 Score

Location: /vulnerabilities/[]/ratings/[]/score
Type: Number

The numerical score of the rating.

6.11.1.5.6 Severity

Location: /vulnerabilities/[]/ratings/[]/severity
Type: String (enum)

Textual representation of the severity that corresponds to the numerical score of the rating.

© Ecma International 2025

Table 289 — 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.

6.11.1.5.7 Method

Location: /vulnerabilities/[]/ratings/[]/method
Type: String (enum)

Specifies the severity or risk scoring methodology or standard used.

Table 290 — 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.

6.11.1.5.8 Vector

Location: /vulnerabilities/[]/ratings/[]/vector
Type: String

Textual representation of the metric values used to score the vulnerability.

6.11.1.5.9 Justification

Location: /vulnerabilities/[]/ratings/[]/justification
Type: String

A reason for rating the vulnerability as it was.

© Ecma International 2025

6.11.1.6 CWEs

Location: /vulnerabilities/[]/cwes
Property: cwes (Optional)
Type: Array (of Integer)
Minimum Value: 1

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 shall be an integer.

Example (Informative)

```
399
```

6.11.1.7 Description

Location: /vulnerabilities/[]/description
Type: String

A description of the vulnerability as provided by the source.

6.11.1.8 Details

Location: /vulnerabilities/[]/detail
Type: String

If available, an in-depth description of the vulnerability as provided by the source organization. Details often include information useful in understanding root cause.

6.11.1.9 Recommendation

Location: /vulnerabilities/[]/recommendation
Type: String

Recommendations of how the vulnerability can be remediated or mitigated.

6.11.1.10 Workarounds

Location: /vulnerabilities/[]/workaround
Type: String

A bypass, usually temporary, of the vulnerability that reduces its likelihood and/or impact. Workarounds often involve changes to configuration or deployments.

6.11.1.11 Proof of Concept

Location: /vulnerabilities/[]/proofOfConcept
Type: Object

Evidence used to reproduce the vulnerability.

© Ecma International 2025

Table 291 — 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.

6.11.1.11.1 Steps to Reproduce

Location: /vulnerabilities/[]/proofOfConcept/reproductionSteps
Type: String

Precise steps to reproduce the vulnerability.

6.11.1.11.2 Environment

Location: /vulnerabilities/[]/proofOfConcept/environment
Type: String

A description of the environment in which reproduction was possible.

6.11.1.11.3 Supporting Material

Location: /vulnerabilities/[]/proofOfConcept/supportingMaterial
Property: supportingMaterial (Optional)
Type: Array

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 shall be an Attachment object.

6.11.1.11.4 Attachment

Location: /vulnerabilities/[]/proofOfConcept/supportingMaterial/[]
Type: Object

Specifies the metadata and content for an attachment.

Table 292 — 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 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.

© Ecma International 2025

6.11.1.11.5 Content-Type

Location: /vulnerabilities/[]/proofOfConcept/supportingMaterial/[]/contentType
Type: String
Default Value: text/plain

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.

Example 1 (Informative)

```
text/plain
```

Example 2 (Informative)

```
application/json
```

Example 3 (Informative)

```
image/png
```

6.11.1.11.6 Encoding

Location: /vulnerabilities/[]/proofOfConcept/supportingMaterial/[]/encoding
Type: String (enum)

Specifies the encoding the text is represented in.

Table 293 — Enumeration of possible values

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

6.11.1.11.7 Attachment Text

Location: /vulnerabilities/[]/proofOfConcept/supportingMaterial/[]/content
Type: String

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

6.11.1.12 Advisories

Location: /vulnerabilities/[]/advisories
Property: advisories (Optional)
Type: Array

Published advisories of the vulnerability if provided. Each item of this array shall be an Advisory object.

© Ecma International 2025

6.11.1.12.1 Advisory

Location: /vulnerabilities/[]/advisories/[]
Type: Object

Title and location where advisory information can be obtained. An advisory is a notification of a threat to a component, service, or system.

Table 294 — Properties for the advisory object

Property	Type	Requirement	Description
title	String	Optional	A name of the advisory.
url	String	Required	Location where the advisory can be obtained.

6.11.1.12.2 Title

Location: /vulnerabilities/[]/advisories/[]/title
Type: String

A name of the advisory.

6.11.1.12.3 URL

Location: /vulnerabilities/[]/advisories/[]/url
Type: String
Format: iri-reference as specified in RFC 3987

Location where the advisory can be obtained.

6.11.1.13 Created

Location: /vulnerabilities/[]/created
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the vulnerability record was created in the vulnerability database.

6.11.1.14 Published

Location: /vulnerabilities/[]/published
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the vulnerability record was first published.

6.11.1.15 Updated

Location: /vulnerabilities/[]/updated
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the vulnerability record was last updated.

© Ecma International 2025

6.11.1.16 Rejected

Location: /vulnerabilities/[]/rejected
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the vulnerability record was rejected (if applicable).

6.11.1.17 Credits

Location: /vulnerabilities/[]/credits
Type: Object

Individuals or organizations credited with the discovery of the vulnerability.

Table 295 — 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.

6.11.1.17.1 Organizations

Location: /vulnerabilities/[]/credits/organizations
Property: organizations (Optional)
Type: Array

The organizations credited with vulnerability discovery.

6.11.1.17.2 Organization

Location: /vulnerabilities/[]/credits/organizations/[]
Type: Object

Table 296 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

© Ecma International 2025

6.11.1.17.3 BOM Reference

Location: /vulnerabilities/[]/credits/organizations/[]/bom-ref
Type: String

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

6.11.1.17.4 Organization Name

Location: /vulnerabilities/[]/credits/organizations/[]/name
Type: String

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.11.1.17.5 Organization Address

Location: /vulnerabilities/[]/credits/organizations/[]/address
Type: Object

The physical address (location) of the organization.

Table 297 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.11.1.17.6 BOM Reference

Location: /vulnerabilities/[]/credits/organizations/[]/address/bom-ref
Type: String

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

© Ecma International 2025

6.11.1.17.7 Country

Location: /vulnerabilities/[]/credits/organizations/[]/address/country
Type: String

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

6.11.1.17.8 Region

Location: /vulnerabilities/[]/credits/organizations/[]/address/region
Type: String

The region or state in the country.

Example (Informative)
```
Texas
```

6.11.1.17.9 Locality

Location: /vulnerabilities/[]/credits/organizations/[]/address/locality
Type: String

The locality or city within the country.

Example (Informative)
```
Austin
```

6.11.1.17.10 Post Office Box Number

Location: /vulnerabilities/[]/credits/organizations/[]/address/postOfficeBoxNumber
Type: String

The post office box number.

Example (Informative)
```
901
```

6.11.1.17.11 Postal Code

Location: /vulnerabilities/[]/credits/organizations/[]/address/postalCode
Type: String

The postal code.

Example (Informative)
```
78758
```


© Ecma International 2025

6.11.1.17.12 Street Address

Location: /vulnerabilities/[]/credits/organizations/[]/address/streetAddress
Type: String

The street address.

Example (Informative)

```
100 Main Street
```

6.11.1.17.13 Organization URL(s)

Location: /vulnerabilities/[]/credits/organizations/[]/url
Property: url (Optional)
Type: Array (of String)
Format: iri-reference as specified in RFC 3987

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

Example (Informative)

```
https://example.com
```

6.11.1.17.14 Organizational Contact

Location: /vulnerabilities/[]/credits/organizations/[]/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.11.1.17.15 Organizational Person

Location: /vulnerabilities/[]/credits/organizations/[]/contact/[]
Type: Object

Table 298 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.11.1.17.16 BOM Reference

Location: /vulnerabilities/[]/credits/organizations/[]/contact/[]/bom-ref
Type: String

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

© Ecma International 2025

6.11.1.17.17 Name

Location: /vulnerabilities/[]/credits/organizations/[]/contact/[]/name
Type: String

The name of a contact.

Example (Informative)
```
Contact name
```

6.11.1.17.18 Email Address

Location: /vulnerabilities/[]/credits/organizations/[]/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.11.1.17.19 Phone

Location: /vulnerabilities/[]/credits/organizations/[]/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.11.1.17.20 Individuals

Location: /vulnerabilities/[]/credits/individuals
Property: individuals (Optional)
Type: Array

The individuals, not associated with organizations, that are credited with vulnerability discovery. Each item of this array shall be an Organizational Person object.

6.11.1.17.21 Organizational Person

Location: /vulnerabilities/[]/credits/individuals/[]
Type: Object

© Ecma International 2025

Table 299 — Properties for the individual object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.11.1.17.22 BOM Reference

Location: /vulnerabilities/[]/credits/individuals/[]/bom-ref
Type: String

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

6.11.1.17.23 Name

Location: /vulnerabilities/[]/credits/individuals/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.11.1.17.24 Email Address

Location: /vulnerabilities/[]/credits/individuals/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

6.11.1.17.25 Phone

Location: /vulnerabilities/[]/credits/individuals/[]/phone
Type: String

The phone number of the contact.

Example (Informative)

```
800-555-1212
```

© Ecma International 2025

6.11.1.18 Tools

Location: /vulnerabilities/[]/tools

The tool(s) used to identify, confirm, or score the vulnerability.

Shall be one of:

1. Tools
2. Tools (legacy)

6.11.1.19 Tools

Type: Object

The tool(s) used to identify, confirm, or score the vulnerability.

Table 300 — Properties for the tools object

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/[].

LEGACY

6.11.1.20 Tools (legacy)

Type: Array

[Deprecated] The tool(s) used to identify, confirm, or score the vulnerability.

6.11.1.21 Components

Location: /vulnerabilities/[]/tools/components
Property: components (Optional)
Type: Array
Uniqueness: All items shall be unique.

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

6.11.1.21.1 Component

Location: /vulnerabilities/[]/tools/components/[]
Type: Object

© Ecma International 2025

6.11.1.22 Services

Location: /vulnerabilities/[]/tools/services
Property: services (Optional)
Type: Array
Uniqueness: All items shall be unique.

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 shall be a Service object.

6.11.1.22.1 Service

Location: /vulnerabilities/[]/tools/services/[]
Type: Object

LEGACY

6.11.1.23 Tools (legacy)

Location: /vulnerabilities/[]/tools
Property: tools
Type: Array

[Deprecated] The tool(s) used to identify, confirm, or score the vulnerability. Each item of this array shall be a Tool object.

6.11.1.23.1 Tool

Location: /vulnerabilities/[]/tools/[]
Type: Object

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

Table 301 — Properties for the tool 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.

© Ecma International 2025

6.11.1.23.2 Tool Vendor

Location: /vulnerabilities/[]/tools/[]/vendor
Type: String

The name of the vendor who created the tool.

6.11.1.23.3 Tool Name

Location: /vulnerabilities/[]/tools/[]/name
Type: String

The name of the tool.

6.11.1.23.4 Tool Version

Location: /vulnerabilities/[]/tools/[]/version
Type: String

The version of the tool.

Example 1 (Informative)
```
9.0.14
```

Example 2 (Informative)
```
v1.33.7
```

Example 3 (Informative)
```
7.0.0-M1
```

Example 4 (Informative)
```
2.0pre1
```

Example 5 (Informative)
```
1.0.0-beta1
```

Example 6 (Informative)
```
0.8.15
```

6.11.1.23.5 Hashes

Location: /vulnerabilities/[]/tools/[]/hashes
Property: hashes (Optional)
Type: Array

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

© Ecma International 2025

6.11.1.23.6 Hash

Location: /vulnerabilities/[]/tools/[]/hashes/[]
Type: Object

Table 302 — Properties for the hash object

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

6.11.1.23.7 Hash Algorithm

Location: /vulnerabilities/[]/tools/[]/hashes/[]/alg
Type: String (enum)

The algorithm that generated the hash value.

Enumeration of possible values:

- MD5
- SHA-1
- SHA-256
- SHA-384
- SHA-512
- SHA3-256
- SHA3-384
- SHA3-512
- BLAKE2b-256
- BLAKE2b-384
- BLAKE2b-512
- BLAKE3
- Streebog-256
- Streebog-512

6.11.1.23.8 Hash Value

Location: /vulnerabilities/[]/tools/[]/hashes/[]/content
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})`

The value of the hash.

Example (Informative)
`3942447fac867ae5cdb3229b658f4d48`

© Ecma International 2025

6.11.1.23.9 External References

Location: /vulnerabilities/[]/tools/[]/externalReferences
Property: externalReferences (Optional)
Type: Array

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 shall be an External Reference object.

6.11.1.23.10 External Reference

Location: /vulnerabilities/[]/tools/[]/externalReferences/[]
Type: Object

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.

6.11.1.24 Impact Analysis

Location: /vulnerabilities/[]/analysis
Type: Object

An assessment of the impact and exploitability of the vulnerability.

Table 303 — 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.

6.11.1.24.1 Impact Analysis State

Location: /vulnerabilities/[]/analysis/state
Type: String (enum)

Declares the current state of an occurrence of a vulnerability, after automated or manual analysis.

© Ecma International 2025

Table 304 — 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).
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.

6.11.1.24.2 Impact Analysis Justification

Location: /vulnerabilities/[]/analysis/justification
Type: String (enum)

The rationale of why the impact analysis state was asserted.

Table 305 — 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.

6.11.1.24.3 Response

Location: /vulnerabilities/[]/analysis/response
Property: response (Optional)
Type: Array (of String) (enum)

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 shall be a string.

© Ecma International 2025

Table 306 — 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.

6.11.1.24.4 Detail

Location: /vulnerabilities/[]/analysis/detail
Type: String

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.

6.11.1.24.5 First Issued

Location: /vulnerabilities/[]/analysis/firstIssued
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the analysis was first issued.

6.11.1.24.6 Last Updated

Location: /vulnerabilities/[]/analysis/lastUpdated
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the analysis was last updated.

6.11.1.25 Affects

Location: /vulnerabilities/[]/affects
Property: affects (Optional)
Type: Array
Uniqueness: All items shall be unique.

The components or services that are affected by the vulnerability.

© Ecma International 2025

6.11.1.25.1 Affect

Location: /vulnerabilities/[]/affects/[]
Type: Object

Table 307 — Properties for the affect 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.

6.11.1.25.2 Reference

Location: /vulnerabilities/[]/affects/[]/ref

References a component or service by the objects bom-ref.

Shall be any of:

1. Ref
2. BOM-Link Element

6.11.1.25.3 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.11.1.25.4 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.11.1.25.5 Versions

Location: /vulnerabilities/[]/affects/[]/versions
Property: versions (Optional)
Type: Array

Zero or more individual versions or range of versions.

6.11.1.25.6 Version

Location: /vulnerabilities/[]/affects/[]/versions/[]
Type: Object

© Ecma International 2025

Table 308 — Properties for the version 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/vers-spec.
status	String	Optional	The vulnerability status for the version or range of versions.

6.11.1.25.7 Version

Location: /vulnerabilities/[]/affects/[]/versions/[]/version
Type: String

A single version of a component or service.

Example 1 (Informative)
```
9.0.14
```

Example 2 (Informative)
```
v1.33.7
```

Example 3 (Informative)
```
7.0.0-M1
```

Example 4 (Informative)
```
2.0pre1
```

Example 5 (Informative)
```
1.0.0-beta1
```

Example 6 (Informative)
```
0.8.15
```

6.11.1.25.8 Version Range

Location: /vulnerabilities/[]/affects/[]/versions/[]/range
Type: String

A version range specified in Package URL Version Range syntax (vers) which is defined at https://github.com/package-url/vers-spec.

Example 1 (Informative)
```
vers:cargo/9.0.14
```

Example 2 (Informative)
```
vers:npm/1.2.3|>=2.0.0|<5.0.0
```

Example 3 (Informative)
```
vers:pypi/0.0.0|0.0.1|0.0.2|0.0.3|1.0|2.0pre1
```


© Ecma International 2025

Example 4 (Informative)

```
ers: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.
```

Example 5 (Informative)

```
vers:gem/>=2.2.0|!= 2.2.1|<2.3.0
```

6.11.1.25.9 Status

Location: /vulnerabilities/[]/affects/[]/versions/[]/status
Type: String (enum)
Default Value: affected

The vulnerability status for the version or range of versions.

Table 309 — 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.

6.11.1.26 Properties

Location: /vulnerabilities/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.11.1.26.1 Lightweight name-value pair

Location: /vulnerabilities/[]/properties/[]
Type: Object

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 310 — Properties for the property 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.

© Ecma International 2025

6.11.1.26.2 Name

Location: /vulnerabilities/[]/properties/[]/name
Type: String

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

6.11.1.26.3 Value

Location: /vulnerabilities/[]/properties/[]/value
Type: String

The value of the property.

6.12 Annotations

Location: /annotations
Property: annotations (Optional)
Type: Array
Uniqueness: All items shall be unique.

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 shall be an Annotations object.

6.12.1 Annotations

Location: /annotations/[]
Type: Object

A comment, note, explanation, or similar textual content which provides additional context to the object(s) being annotated.

Table 311 — Properties for the annotation object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the annotation elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. 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).

© Ecma International 2025

6.12.1.1 BOM Reference

Location: /annotations/[]/bom-ref
Type: String

An identifier which can be used to reference the annotation elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. Value should not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

6.12.1.2 Subjects

Location: /annotations/[]/subjects
Property: subjects (Required)
Type: Array
Uniqueness: All items shall be unique.

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.

6.12.1.2.1 Subject

Location: /annotations/[]/subjects/[]

Shall be any of:

1. Ref
2. BOM-Link Element

6.12.1.2.2 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.12.1.2.3 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.12.1.3 Annotator

Location: /annotations/[]/annotator
Type: Object

The organization, person, component, or service which created the textual content of the annotation.

© Ecma International 2025

Table 312 — 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	Array	Optional	The tool or component that created the annotation.
service	Object	Optional	The service that created the annotation.

6.12.1.3.1 Organization

Location: /annotations/[]/annotator/organization
Type: Object

The organization that created the annotation.

Table 313 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.12.1.3.2 BOM Reference

Location: /annotations/[]/annotator/organization/bom-ref
Type: String

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

6.12.1.3.3 Organization Name

Location: /annotations/[]/annotator/organization/name
Type: String

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.12.1.3.4 Organization Address

Location: /annotations/[]/annotator/organization/address
Type: Object

The physical address (location) of the organization.

© Ecma International 2025

Table 314 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.12.1.3.5 BOM Reference

Location: /annotations/[]/annotator/organization/address/bom-ref
Type: String

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

6.12.1.3.6 Country

Location: /annotations/[]/annotator/organization/address/country
Type: String

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

6.12.1.3.7 Region

Location: /annotations/[]/annotator/organization/address/region
Type: String

The region or state in the country.

Example (Informative)

```
Texas
```

6.12.1.3.8 Locality

Location: /annotations/[]/annotator/organization/address/locality
Type: String

The locality or city within the country.

Example (Informative)

```
Austin
```

© Ecma International 2025

6.12.1.3.9 Post Office Box Number

Location: /annotations/[]/annotator/organization/address/postOfficeBoxNumber
Type: String

The post office box number.

Example (Informative)
```
901
```

6.12.1.3.10 Postal Code

Location: /annotations/[]/annotator/organization/address/postalCode
Type: String

The postal code.

Example (Informative)
```
78758
```

6.12.1.3.11 Street Address

Location: /annotations/[]/annotator/organization/address/streetAddress
Type: String

The street address.

Example (Informative)
```
100 Main Street
```

6.12.1.3.12 Organization URL(s)

Location: /annotations/[]/annotator/organization/url
Property: url (Optional)
Type: Array (of String)
Format: iri-reference as specified in RFC 3987

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

Example (Informative)
```
https://example.com
```

6.12.1.3.13 Organizational Contact

Location: /annotations/[]/annotator/organization/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.12.1.3.14 Organizational Person

Location: /annotations/[]/annotator/organization/contact/[]
Type: Object

© Ecma International 2025

Table 315 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.12.1.3.15 BOM Reference

Location: /annotations/[]/annotator/organization/contact/[]/bom-ref
Type: String

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

6.12.1.3.16 Name

Location: /annotations/[]/annotator/organization/contact/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.12.1.3.17 Email Address

Location: /annotations/[]/annotator/organization/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

6.12.1.3.18 Phone

Location: /annotations/[]/annotator/organization/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)

```
800-555-1212
```

© Ecma International 2025

6.12.1.3.19 Organizational Person

Location: /annotations/[]/annotator/individual
Type: Object

The person that created the annotation.

Table 316 — Properties for the individual object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.12.1.3.20 BOM Reference

Location: /annotations/[]/annotator/individual/bom-ref
Type: String

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

6.12.1.3.21 Name

Location: /annotations/[]/annotator/individual/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.12.1.3.22 Email Address

Location: /annotations/[]/annotator/individual/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```


© Ecma International 2025

6.12.1.3.23 Phone

Location: /annotations/[]/annotator/individual/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.12.1.3.24 Component

Location: /annotations/[]/annotator/component
Type: Object

The tool or component that created the annotation.

6.12.1.3.25 Service

Location: /annotations/[]/annotator/service
Type: Object

The service that created the annotation.

6.12.1.4 Timestamp

Location: /annotations/[]/timestamp
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the annotation was created.

6.12.1.5 Text

Location: /annotations/[]/text
Type: String

The textual content of the annotation.

6.12.1.6 Signature

Location: /annotations/[]/signature
Type: Object

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.

© Ecma International 2025

6.13 Formulation

Location: /formulation
Property: formulation (Optional)
Type: Array
Uniqueness: All items shall be unique.

Describes the formulation of any referencable object within the BOM, including components, services, metadata, declarations, or the BOM itself. This may encompass how the object was created, assembled, deployed, tested, certified, or otherwise brought into its present form. Common examples include software build pipelines, deployment processes, AI/ML model training, cryptographic key generation or certification, and third-party audits. Processes are modelled using declared and observed formulas, composed of workflows, tasks, and individual steps. Each item of this array shall be a Formula object.

6.13.1 Formula

Location: /formulation/[]
Type: Object

Describes workflows and resources that captures rules and other aspects of how the associated BOM component or service was formed.

Table 317 — Properties for the formulation object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the formula elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. 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.

6.13.1.1 BOM Reference

Location: /formulation/[]/bom-ref
Type: String

An identifier which can be used to reference the formula elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. Value should not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

© Ecma International 2025

6.13.1.2 Components

Location: /formulation/[]/components
Property: components (Optional)
Type: Array
Uniqueness: All items shall be unique.

Transient components that are used in tasks that constitute one or more of this formula's workflows Each item of this array shall be a Component object.

6.13.1.2.1 Component

Location: /formulation/[]/components/[]
Type: Object

6.13.1.3 Services

Location: /formulation/[]/services
Property: services (Optional)
Type: Array
Uniqueness: All items shall be unique.

Transient services that are used in tasks that constitute one or more of this formula's workflows Each item of this array shall be a Service object.

6.13.1.3.1 Service

Location: /formulation/[]/services/[]
Type: Object

6.13.1.4 Workflows

Location: /formulation/[]/workflows
Property: workflows (Optional)
Type: Array
Uniqueness: All items shall be unique.

List of workflows that can be declared to accomplish specific orchestrated goals and independently triggered. Each item of this array shall be a Workflow object.

6.13.1.4.1 Workflow

Location: /formulation/[]/workflows/[]
Type: Object

A specialized orchestration task.

© Ecma International 2025

Table 318 — Properties for the workflow object

Property	Type	Requirement	Description
bom-ref	String	Required	An identifier which can be used to reference the workflow elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. 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.
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.

6.13.1.4.2 BOM Reference

Location: /formulation/[]/workflows/[]/bom-ref
Type: String

An identifier which can be used to reference the workflow elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. Value should not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

© Ecma International 2025

6.13.1.4.3 Unique Identifier (UID)

Location: /formulation/[]/workflows/[]/uid
Type: String

The unique identifier for the resource instance within its deployment context.

6.13.1.4.4 Name

Location: /formulation/[]/workflows/[]/name
Type: String

The name of the resource instance.

6.13.1.4.5 Description

Location: /formulation/[]/workflows/[]/description
Type: String

A description of the resource instance.

6.13.1.4.6 Resource references

Location: /formulation/[]/workflows/[]/resourceReferences
Property: resourceReferences (Optional)
Type: Array
Uniqueness: All items shall be unique.

References to component or service resources that are used to realize the resource instance. Each item of this array shall be a Resource reference choice object.

6.13.1.4.7 Resource reference choice

Location: /formulation/[]/workflows/[]/resourceReferences/[]
Type: Object

A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Table 319 — Properties for the resourceReference object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.8 BOM Reference

Location: /formulation/[]/workflows/[]/resourceReferences/[]/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

© Ecma International 2025

6.13.1.4.9 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.10 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.11 External reference

Location: /formulation/[]/workflows/[]/resourceReferences/[]/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.12 Tasks

Location: /formulation/[]/workflows/[]/tasks
Property: tasks (Optional)
Type: Array
Uniqueness: All items shall be unique.

The tasks that comprise the workflow. Each item of this array shall be a Task object.

6.13.1.4.13 Task

Location: /formulation/[]/workflows/[]/tasks/[]
Type: Object

Describes the inputs, sequence of steps and resources used to accomplish a task and its output.

Table 320 — Properties for the task object

Property	Type	Requirement	Description
bom-ref	String	Required	An identifier which can be used to reference the task elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. 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.

© Ecma International 2025

Table 320 — Properties for the task object *(continued)*

Property	Type	Requirement	Description
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 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.

6.13.1.4.14 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/bom-ref
Type: String

An identifier which can be used to reference the task elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. Value should not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

6.13.1.4.15 Unique Identifier (UID)

Location: /formulation/[]/workflows/[]/tasks/[]/uid
Type: String

The unique identifier for the resource instance within its deployment context.

6.13.1.4.16 Name

Location: /formulation/[]/workflows/[]/tasks/[]/name
Type: String

The name of the resource instance.

© Ecma International 2025

6.13.1.4.17 Description

Location: /formulation/[]/workflows/[]/tasks/[]/description
Type: String

A description of the resource instance.

6.13.1.4.18 Resource references

Location: /formulation/[]/workflows/[]/tasks/[]/resourceReferences
Property: resourceReferences (Optional)
Type: Array
Uniqueness: All items shall be unique.

References to component or service resources that are used to realize the resource instance. Each item of this array shall be a Resource reference choice object.

6.13.1.4.19 Resource reference choice

Location: /formulation/[]/workflows/[]/tasks/[]/resourceReferences/[]
Type: Object

A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Table 321 — Properties for the resourceReference object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.20 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/resourceReferences/[]/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.21 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.22 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

© Ecma International 2025

6.13.1.4.23 External reference

Location: /formulation/[]/workflows/[]/tasks/[]/resourceReferences/[]/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.24 Task types

Location: /formulation/[]/workflows/[]/tasks/[]/taskTypes
Property: taskTypes (Required)
Type: Array (of String) (enum)

Indicates the types of activities performed by the set of workflow tasks. Each item of this array shall be a string.

Table 322 — 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.

6.13.1.4.25 Trigger

Location: /formulation/[]/workflows/[]/tasks/[]/trigger
Type: Object

The trigger that initiated the task.

© Ecma International 2025

Table 323 — Properties for the trigger object

Property	Type	Requirement	Description
bom-ref	String	Required	An identifier which can be used to reference the trigger elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. 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.

6.13.1.4.26 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/bom-ref
Type: String

An identifier which can be used to reference the trigger elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. Value should not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

6.13.1.4.27 Unique Identifier (UID)

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/uid
Type: String

The unique identifier for the resource instance within its deployment context.

© Ecma International 2025

6.13.1.4.28 Name

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/name
Type: String

The name of the resource instance.

6.13.1.4.29 Description

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/description
Type: String

A description of the resource instance.

6.13.1.4.30 Resource references

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/resourceReferences
Property: resourceReferences (Optional)
Type: Array
Uniqueness: All items shall be unique.

References to component or service resources that are used to realize the resource instance. Each item of this array shall be a Resource reference choice object.

6.13.1.4.31 Resource reference choice

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/resourceReferences/[]
Type: Object

A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Table 324 — Properties for the resourceReference object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.32 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/resourceReferences/[]/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.33 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

© Ecma International 2025

6.13.1.4.34 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.35 External reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/resourceReferences/[]/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.36 Type

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/type
Type: String (enum)

The source type of event which caused the trigger to fire.

Enumeration of possible values:

- manual
- api
- webhook
- scheduled

6.13.1.4.37 Event

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event
Type: Object

The event data that caused the associated trigger to activate.

Table 325 — 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	Encoding of the raw event data.
source	Array	Optional	References the component or service that was the source of the event.

© Ecma International 2025

Table 325 — Properties for the event object *(continued)*

Property	Type	Requirement	Description
target	Array	Optional	References the component or service that was the target of the event.
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.

6.13.1.4.38 Unique Identifier (UID)

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/uid
Type: String

The unique identifier of the event.

6.13.1.4.39 Description

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/description
Type: String

A description of the event.

6.13.1.4.40 Time Received

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/timeReceived
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the event was received.

6.13.1.4.41 Data

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/data
Type: Object

Encoding of the raw event data.

Table 326 — 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 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.

© Ecma International 2025

6.13.1.4.42 Content-Type

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/data/contentType
Type: String
Default Value: text/plain

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.

Example 1 (Informative)
```
text/plain
```

Example 2 (Informative)
```
application/json
```

Example 3 (Informative)
```
image/png
```

6.13.1.4.43 Encoding

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/data/encoding
Type: String (enum)

Specifies the encoding the text is represented in.

Table 327 — Enumeration of possible values

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

6.13.1.4.44 Attachment Text

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/data/content
Type: String

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

6.13.1.4.45 Source

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/source
Type: Object

References the component or service that was the source of the event.

Table 328 — Properties for the source object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

© Ecma International 2025

6.13.1.4.46 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/source/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.47 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.48 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.49 External reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/source/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.50 Target

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/target
Type: Object

References the component or service that was the target of the event.

Table 329 — Properties for the target object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.51 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/target/ref

References an object by its bom-ref attribute.

© Ecma International 2025

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.52 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.53 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.54 External reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/target/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.55 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.13.1.4.56 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/properties/[]
Type: Object

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.

© Ecma International 2025

Table 330 — Properties for the property 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.

6.13.1.4.57 Name

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/properties/[]/name
Type: String

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

6.13.1.4.58 Value

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/event/properties/[]/value
Type: String

The value of the property.

6.13.1.4.59 Conditions

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/conditions
Property: conditions (Optional)
Type: Array
Uniqueness: All items shall be unique.

A list of conditions used to determine if a trigger should be activated. Each item of this array shall be a Condition object.

6.13.1.4.60 Condition

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/conditions/[]
Type: Object

A condition that was used to determine a trigger should be activated.

Table 331 — Properties for the condition 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.

© Ecma International 2025

6.13.1.4.61 Description

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/conditions/[]/description
Type: String

Describes the set of conditions which cause the trigger to activate.

6.13.1.4.62 Expression

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/conditions/[]/expression
Type: String

The logical expression that was evaluated that determined the trigger should be fired.

6.13.1.4.63 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/conditions/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.13.1.4.64 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/conditions/[]/properties/[]
Type: Object

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 332 — Properties for the property 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.

6.13.1.4.65 Name

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/conditions/[]/properties/[]/name
Type: String

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

© Ecma International 2025

6.13.1.4.66 Value

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/conditions/[]/properties/[]/value
Type: String

The value of the property.

6.13.1.4.67 Time activated

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/timeActivated
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the trigger was activated.

6.13.1.4.68 Inputs

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs
Property: inputs (Optional)
Type: Array
Uniqueness: All items shall be unique.

Represents resources and data brought into a task at runtime by executor or task commands Each item of this array shall be an Input type object.

6.13.1.4.69 Input type

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]
Type: Object

Type that represents various input data types and formats.

Table 333 — Properties for the input object

Property	Type	Requirement	Description
source	Array	Optional	A reference to the component or service that provided the input to the task (e.g., reference to a service with data flow value of **`inbound`**).
target	Array	Optional	A reference to the component or service that received or stored the input if not the task itself (e.g., a local, named storage workspace).
resource	Array	Optional	A reference to an independent resource provided as an input to a task by the workflow runtime.
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.

© Ecma International 2025

Table 333 — Properties for the input object *(continued)*

Property	Type	Requirement	Description
data	Object	Optional	Inputs that have the form of data.
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.

6.13.1.4.70 Source

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/source
Type: Object

A reference to the component or service that provided the input to the task (e.g., reference to a service with data flow value of **`inbound`**).

Example 1 (Informative)

```
source code repository
```

Example 2 (Informative)

```
database
```

Table 334 — Properties for the source object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

© Ecma International 2025

6.13.1.4.71 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/source/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.72 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.73 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.74 External reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/source/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.75 Target

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/target
Type: Object

A reference to the component or service that received or stored the input if not the task itself (e.g., a local, named storage workspace).

Example 1 (Informative)

```
workspace
```

Example 2 (Informative)

```
directory
```

Table 335 — Properties for the target object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

© Ecma International 2025

6.13.1.4.76 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/target/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.77 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.78 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.79 External reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/target/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.80 Resource

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/resource
Type: Object

A reference to an independent resource provided as an input to a task by the workflow runtime.

Example 1 (Informative)

```
a reference to a configuration file in a repository (i.e., a bom-ref)
```

Example 2 (Informative)

```
a reference to a scanning service used in a task (i.e., a bom-ref)
```

Table 336 — Properties for the resource object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

© Ecma International 2025

6.13.1.4.81 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/resource/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.82 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.83 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.84 External reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/resource/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.85 Parameters

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/parameters
Property: parameters (Optional)
Type: Array
Uniqueness: All items shall be unique.

Inputs that have the form of parameters with names and values. Each item of this array shall be a Parameter object.

6.13.1.4.86 Parameter

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/parameters/[]
Type: Object

A representation of a functional parameter.

© Ecma International 2025

Table 337 — Properties for the parameter 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.

6.13.1.4.87 Name

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/parameters/[]/name
Type: String

The name of the parameter.

6.13.1.4.88 Value

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/parameters/[]/value
Type: String

The value of the parameter.

6.13.1.4.89 Data type

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/parameters/[]/dataType
Type: String

The data type of the parameter.

6.13.1.4.90 Environment variables

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/environmentVars
Property: environmentVars (Optional)
Type: Array
Uniqueness: All items shall be unique.

Inputs that have the form of parameters with names and values.

6.13.1.4.91 EnvironmentVar

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/environmentVars/[]

Shall be one of:

1. Lightweight name-value pair
2. String-Based Environment Variables

6.13.1.4.92 Lightweight name-value pair

Type: Object

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.

© Ecma International 2025

Table 338 — Properties for the lightweight name-value pair 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.

6.13.1.4.93 String-Based Environment Variables

Type: String

In addition to the more common key–value pair format, some environment variables may consist of a single string without an explicit value assignment. These string-based environment variables typically act as flags or signals to software, indicating that a feature should be enabled, a mode should be activated, or a specific condition is present. Their presence alone conveys meaning.

6.13.1.4.94 Name

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/environmentVars/[]/name
Property: name (Required)
Type: String

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

6.13.1.4.95 Value

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/environmentVars/[]/value
Property: value (Optional)
Type: String

The value of the property.

6.13.1.4.96 Data

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/data
Type: Object

Inputs that have the form of data.

Table 339 — 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 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.

© Ecma International 2025

6.13.1.4.97 Content-Type

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/data/contentType
Type: String
Default Value: text/plain

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.

Example 1 (Informative)

```
text/plain
```

Example 2 (Informative)

```
application/json
```

Example 3 (Informative)

```
image/png
```

6.13.1.4.98 Encoding

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/data/encoding
Type: String (enum)

Specifies the encoding the text is represented in.

Table 340 — Enumeration of possible values

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

6.13.1.4.99 Attachment Text

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/data/content
Type: String

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

6.13.1.4.100 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

© Ecma International 2025

6.13.1.4.101 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/properties/[]
Type: Object

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 the property 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.

6.13.1.4.102 Name

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/properties/[]/name
Type: String

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

6.13.1.4.103 Value

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/inputs/[]/properties/[]/value
Type: String

The value of the property.

6.13.1.4.104 Outputs

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs
Property: outputs (Optional)
Type: Array
Uniqueness: All items shall be unique.

Represents resources and data output from a task at runtime by executor or task commands

6.13.1.4.105 Output

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]
Type: Object

© Ecma International 2025

Table 342 — Properties for the output object

Property	Type	Requirement	Description
type	String	Optional	Describes the type of data output.
source	Array	Optional	Component or service that generated or provided the output from the task (e.g., a build tool).
target	Array	Optional	Component or service that received the output from the task (e.g., reference to an artifactory service with data flow value of **`outbound`**).
resource	Array	Optional	A reference to an independent resource generated as output by the task.
data	Object	Optional	Outputs that have the form of data.
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.

6.13.1.4.106 Type

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/type
Type: String (enum)

Describes the type of data output.

Enumeration of possible values:

- artifact
- attestation
- log
- evidence
- metrics
- other

6.13.1.4.107 Source

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/source
Type: Object

Component or service that generated or provided the output from the task (e.g., a build tool).

Table 343 — Properties for the source object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

© Ecma International 2025

6.13.1.4.108 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/source/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.109 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.110 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.111 External reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/source/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.112 Target

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/target
Type: Object

Component or service that received the output from the task (e.g., reference to an artifactory service with data flow value of **`outbound`**).

Example (Informative)

```
a log file described as an `externalReference` within its target domain.
```

Table 344 — Properties for the target object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

© Ecma International 2025

6.13.1.4.113 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/target/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.114 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.115 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.116 External reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/target/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.117 Resource

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/resource
Type: Object

A reference to an independent resource generated as output by the task.

Example 1 (Informative)
```
configuration file
```

Example 2 (Informative)
```
source code
```

Example 3 (Informative)
```
scanning service
```


© Ecma International 2025

Table 345 — Properties for the resource object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.118 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/resource/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.119 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.120 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.121 External reference

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/resource/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.122 Data

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/data
Type: Object

Outputs that have the form of data.

© Ecma International 2025

Table 346 — 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 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.

6.13.1.4.123 Content-Type

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/data/contentType
Type: String
Default Value: text/plain

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.

Example 1 (Informative)

```
text/plain
```

Example 2 (Informative)

```
application/json
```

Example 3 (Informative)

```
image/png
```

6.13.1.4.124 Encoding

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/data/encoding
Type: String (enum)

Specifies the encoding the text is represented in.

Table 347 — Enumeration of possible values

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

6.13.1.4.125 Attachment Text

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/data/content
Type: String

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

© Ecma International 2025

6.13.1.4.126 Environment variables

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/environmentVars
Property: environmentVars (Optional)
Type: Array
Uniqueness: All items shall be unique.

Outputs that have the form of environment variables.

6.13.1.4.127 EnvironmentVar

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/environmentVars/[]

Shall be one of:

1. Lightweight name-value pair
2. String-Based Environment Variables

6.13.1.4.128 Lightweight name-value pair

Type: Object

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 348 — Properties for the lightweight name-value pair 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.

6.13.1.4.129 String-Based Environment Variables

Type: String

In addition to the more common key–value pair format, some environment variables may consist of a single string without an explicit value assignment. These string-based environment variables typically act as flags or signals to software, indicating that a feature should be enabled, a mode should be activated, or a specific condition is present. Their presence alone conveys meaning.

6.13.1.4.130 Name

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/environmentVars/[]/name
Property: name (Required)
Type: String

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

© Ecma International 2025

6.13.1.4.131 Value

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/environmentVars/[]/value
Property: value (Optional)
Type: String

The value of the property.

6.13.1.4.132 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.13.1.4.133 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/properties/[]
Type: Object

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 349 — Properties for the property 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.

6.13.1.4.134 Name

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/properties/[]/name
Type: String

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

6.13.1.4.135 Value

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/outputs/[]/properties/[]/value
Type: String

The value of the property.

© Ecma International 2025

6.13.1.4.136 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.13.1.4.137 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/properties/[]
Type: Object

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 350 — Properties for the property 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.

6.13.1.4.138 Name

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/properties/[]/name
Type: String

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

6.13.1.4.139 Value

Location: /formulation/[]/workflows/[]/tasks/[]/trigger/properties/[]/value
Type: String

The value of the property.

6.13.1.4.140 Steps

Location: /formulation/[]/workflows/[]/tasks/[]/steps
Property: steps (Optional)
Type: Array
Uniqueness: All items shall be unique.

The sequence of steps for the task.

© Ecma International 2025

6.13.1.4.141 Step

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]
Type: Object

Executes specific commands or tools in order to accomplish its owning task as part of a sequence.

Table 351 — Properties for the step 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.
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.

6.13.1.4.142 Name

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/name
Type: String

A name for the step.

6.13.1.4.143 Description

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/description
Type: String

A description of the step.

6.13.1.4.144 Commands

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/commands
Property: commands (Optional)
Type: Array

Ordered list of commands or directives for the step

6.13.1.4.145 Command

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/commands/[]
Type: Object

© Ecma International 2025

Table 352 — Properties for the command 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.

6.13.1.4.146 Executed

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/commands/[]/executed
Type: String

A text representation of the executed command.

6.13.1.4.147 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/commands/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.13.1.4.148 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/commands/[]/properties/[]
Type: Object

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 property 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.

© Ecma International 2025

6.13.1.4.149 Name

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/commands/[]/properties/[]/name
Type: String

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

6.13.1.4.150 Value

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/commands/[]/properties/[]/value
Type: String

The value of the property.

6.13.1.4.151 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.13.1.4.152 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/properties/[]
Type: Object

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 354 — Properties for the property 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.

6.13.1.4.153 Name

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/properties/[]/name
Type: String

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

© Ecma International 2025

6.13.1.4.154 Value

Location: /formulation/[]/workflows/[]/tasks/[]/steps/[]/properties/[]/value
Type: String

The value of the property.

6.13.1.4.155 Inputs

Location: /formulation/[]/workflows/[]/tasks/[]/inputs
Property: inputs (Optional)
Type: Array
Uniqueness: All items shall be unique.

Represents resources and data brought into a task at runtime by executor or task commands Each item of this array shall be an Input type object.

6.13.1.4.156 Input type

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]
Type: Object

Type that represents various input data types and formats.

Table 355 — Properties for the input object

Property	Type	Requirement	Description
source	Array	Optional	A reference to the component or service that provided the input to the task (e.g., reference to a service with data flow value of **`inbound`**).
target	Array	Optional	A reference to the component or service that received or stored the input if not the task itself (e.g., a local, named storage workspace).
resource	Array	Optional	A reference to an independent resource provided as an input to a task by the workflow runtime.
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	Inputs that have the form of data.
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.

© Ecma International 2025

6.13.1.4.157 Source

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/source
Type: Object

A reference to the component or service that provided the input to the task (e.g., reference to a service with data flow value of **`inbound`**).

Example 1 (Informative)
```
source code repository
```

Example 2 (Informative)
```
database
```

Table 356 — Properties for the source object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.158 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/source/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.159 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.160 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+
```

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.161 External reference

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/source/externalReference
Type: Object

Reference to an externally accessible resource.

© Ecma International 2025

6.13.1.4.162 Target

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/target
Type: Object

A reference to the component or service that received or stored the input if not the task itself (e.g., a local, named storage workspace).

Example 1 (Informative)

```
workspace
```

Example 2 (Informative)

```
directory
```

Table 357 — Properties for the target object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.163 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/target/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.164 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.165 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.166 External reference

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/target/externalReference
Type: Object

Reference to an externally accessible resource.

© Ecma International 2025

6.13.1.4.167 Resource

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/resource
Type: Object

A reference to an independent resource provided as an input to a task by the workflow runtime.

Example 1 (Informative)
```
a reference to a configuration file in a repository (i.e., a bom-ref)
```

Example 2 (Informative)
```
a reference to a scanning service used in a task (i.e., a bom-ref)
```

Table 358 — Properties for the resource object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.168 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/resource/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.169 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.170 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+
```

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.171 External reference

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/resource/externalReference
Type: Object

Reference to an externally accessible resource.

© Ecma International 2025

6.13.1.4.172 Parameters

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/parameters
Property: parameters (Optional)
Type: Array
Uniqueness: All items shall be unique.

Inputs that have the form of parameters with names and values. Each item of this array shall be a Parameter object.

6.13.1.4.173 Parameter

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/parameters/[]
Type: Object

A representation of a functional parameter.

Table 359 — Properties for the parameter 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.

6.13.1.4.174 Name

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/parameters/[]/name
Type: String

The name of the parameter.

6.13.1.4.175 Value

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/parameters/[]/value
Type: String

The value of the parameter.

6.13.1.4.176 Data type

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/parameters/[]/dataType
Type: String

The data type of the parameter.

6.13.1.4.177 Environment variables

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/environmentVars
Property: environmentVars (Optional)
Type: Array
Uniqueness: All items shall be unique.

Inputs that have the form of parameters with names and values.

© Ecma International 2025

6.13.1.4.178 EnvironmentVar

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/environmentVars/[]

Shall be one of:

1. Lightweight name-value pair
2. String-Based Environment Variables

6.13.1.4.179 Lightweight name-value pair

Type: Object

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 360 — Properties for the lightweight name-value pair 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.

6.13.1.4.180 String-Based Environment Variables

Type: String

In addition to the more common key–value pair format, some environment variables may consist of a single string without an explicit value assignment. These string-based environment variables typically act as flags or signals to software, indicating that a feature should be enabled, a mode should be activated, or a specific condition is present. Their presence alone conveys meaning.

6.13.1.4.181 Name

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/environmentVars/[]/name
Property: name (Required)
Type: String

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

6.13.1.4.182 Value

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/environmentVars/[]/value
Property: value (Optional)
Type: String

The value of the property.

© Ecma International 2025

6.13.1.4.183 Data

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/data
Type: Object

Inputs that have the form of data.

Table 361 — 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 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.

6.13.1.4.184 Content-Type

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/data/contentType
Type: String
Default Value: text/plain

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.

Example 1 (Informative)

```
text/plain
```

Example 2 (Informative)

```
application/json
```

Example 3 (Informative)

```
image/png
```

6.13.1.4.185 Encoding

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/data/encoding
Type: String (enum)

Specifies the encoding the text is represented in.

Table 362 — Enumeration of possible values

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

© Ecma International 2025

6.13.1.4.186 Attachment Text

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/data/content
Type: String

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

6.13.1.4.187 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.13.1.4.188 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/properties/[]
Type: Object

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 the property 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.

6.13.1.4.189 Name

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/properties/[]/name
Type: String

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

6.13.1.4.190 Value

Location: /formulation/[]/workflows/[]/tasks/[]/inputs/[]/properties/[]/value
Type: String

The value of the property.

© Ecma International 2025

6.13.1.4.191 Outputs

Location: /formulation/[]/workflows/[]/tasks/[]/outputs
Property: outputs (Optional)
Type: Array
Uniqueness: All items shall be unique.

Represents resources and data output from a task at runtime by executor or task commands

6.13.1.4.192 Output

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]
Type: Object

Table 364 — Properties for the output object

Property	Type	Requirement	Description
type	String	Optional	Describes the type of data output.
source	Array	Optional	Component or service that generated or provided the output from the task (e.g., a build tool).
target	Array	Optional	Component or service that received the output from the task (e.g., reference to an artifactory service with data flow value of **`outbound`**).
resource	Array	Optional	A reference to an independent resource generated as output by the task.
data	Object	Optional	Outputs that have the form of data.
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.

6.13.1.4.193 Type

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/type
Type: String (enum)

Describes the type of data output.

Enumeration of possible values:

- artifact
- attestation
- log
- evidence
- metrics
- other

© Ecma International 2025

6.13.1.4.194 Source

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/source
Type: Object

Component or service that generated or provided the output from the task (e.g., a build tool).

Table 365 — Properties for the source object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.195 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/source/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.196 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.197 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.198 External reference

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/source/externalReference
Type: Object

Reference to an externally accessible resource.

© Ecma International 2025

6.13.1.4.199 Target

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/target
Type: Object

Component or service that received the output from the task (e.g., reference to an artifactory service with data flow value of **`outbound`**).

Example (Informative)

```
a log file described as an `externalReference` within its target domain.
```

Table 366 — Properties for the target object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.200 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/target/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.201 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.202 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.203 External reference

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/target/externalReference
Type: Object

Reference to an externally accessible resource.

© Ecma International 2025

6.13.1.4.204 Resource

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/resource
Type: Object

A reference to an independent resource generated as output by the task.

Example 1 (Informative)
```
configuration file
```

Example 2 (Informative)
```
source code
```

Example 3 (Informative)
```
scanning service
```

Table 367 — Properties for the resource object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.205 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/resource/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.206 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.207 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+
```

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

© Ecma International 2025

6.13.1.4.208 External reference

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/resource/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.209 Data

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/data
Type: Object

Outputs that have the form of data.

Table 368 — 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 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.

6.13.1.4.210 Content-Type

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/data/contentType
Type: String
Default Value: text/plain

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.

Example 1 (Informative)

```
text/plain
```

Example 2 (Informative)

```
application/json
```

Example 3 (Informative)

```
image/png
```

6.13.1.4.211 Encoding

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/data/encoding
Type: String (enum)

Specifies the encoding the text is represented in.

© Ecma International 2025

Table 369 — Enumeration of possible values

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

6.13.1.4.212 Attachment Text

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/data/content
Type: String

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

6.13.1.4.213 Environment variables

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/environmentVars
Property: environmentVars (Optional)
Type: Array
Uniqueness: All items shall be unique.

Outputs that have the form of environment variables.

6.13.1.4.214 EnvironmentVar

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/environmentVars/[]

Shall be one of:

1. Lightweight name-value pair
2. String-Based Environment Variables

6.13.1.4.215 Lightweight name-value pair

Type: Object

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 370 — Properties for the lightweight name-value pair 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.

6.13.1.4.216 String-Based Environment Variables

Type: String

In addition to the more common key–value pair format, some environment variables may consist of a single string without an explicit value assignment. These string-based environment variables typically act as flags or signals

© Ecma International 2025

to software, indicating that a feature should be enabled, a mode should be activated, or a specific condition is present. Their presence alone conveys meaning.

6.13.1.4.217 Name

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/environmentVars/[]/name
Property: name (Required)
Type: String

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

6.13.1.4.218 Value

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/environmentVars/[]/value
Property: value (Optional)
Type: String

The value of the property.

6.13.1.4.219 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.13.1.4.220 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/properties/[]
Type: Object

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 371 — Properties for the property 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.

6.13.1.4.221 Name

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/properties/[]/name
Type: String

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

© Ecma International 2025

6.13.1.4.222 Value

Location: /formulation/[]/workflows/[]/tasks/[]/outputs/[]/properties/[]/value
Type: String

The value of the property.

6.13.1.4.223 Time start

Location: /formulation/[]/workflows/[]/tasks/[]/timeStart
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the task started.

6.13.1.4.224 Time end

Location: /formulation/[]/workflows/[]/tasks/[]/timeEnd
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the task ended.

6.13.1.4.225 Workspaces

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces
Property: workspaces (Optional)
Type: Array
Uniqueness: All items shall be unique.

A set of named filesystem or data resource shareable by workflow tasks. Each item of this array shall be a Workspace object.

6.13.1.4.226 Workspace

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]
Type: Object

A named filesystem or data resource shareable by workflow tasks.

Table 372 — Properties for the workspace object

Property	Type	Requirement	Description
bom-ref	String	Required	An identifier which can be used to reference the workspace elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. 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.

© Ecma International 2025

Table 372 — Properties for the workspace object *(continued)*

Property	Type	Requirement	Description
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.

6.13.1.4.227 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/bom-ref
Type: String

An identifier which can be used to reference the workspace elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. Value should not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

6.13.1.4.228 Unique Identifier (UID)

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/uid
Type: String

The unique identifier for the resource instance within its deployment context.

6.13.1.4.229 Name

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/name
Type: String

The name of the resource instance.

© Ecma International 2025

6.13.1.4.230 Aliases

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/aliases
Property: aliases (Optional)
Type: Array (of String)

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 shall be a string.

6.13.1.4.231 Description

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/description
Type: String

A description of the resource instance.

6.13.1.4.232 Resource references

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/resourceReferences
Property: resourceReferences (Optional)
Type: Array
Uniqueness: All items shall be unique.

References to component or service resources that are used to realize the resource instance. Each item of this array shall be a Resource reference choice object.

6.13.1.4.233 Resource reference choice

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/resourceReferences/[]
Type: Object

A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Table 373 — Properties for the resourceReference object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.234 BOM Reference

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/resourceReferences/[]/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.235 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

© Ecma International 2025

6.13.1.4.236 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.237 External reference

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/resourceReferences/[]/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.238 Access mode

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/accessMode
Type: String (enum)

Describes the read-write access control for the workspace relative to the owning resource instance.

Enumeration of possible values:

- read-only
- read-write
- read-write-once
- write-once
- write-only

6.13.1.4.239 Mount path

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/mountPath
Type: String

A path to a location on disk where the workspace will be available to the associated task's steps.

6.13.1.4.240 Managed data type

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/managedDataType
Type: String

The name of a domain-specific data type the workspace represents.

Example 1 (Informative)
```
ConfigMap
```

Example 2 (Informative)
```
Secret
```

© Ecma International 2025

6.13.1.4.241 Volume request

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volumeRequest
Type: String

Identifies the reference to the request for a specific volume type and parameters.

Example (Informative)

```
a kubernetes Persistent Volume Claim (PVC) name
```

6.13.1.4.242 Volume

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume
Type: Object

Information about the actual volume instance allocated to the workspace.

Example (Informative)

```
see https://kubernetes.io/docs/concepts/storage/persistent-volumes/
```

Table 374 — 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.

6.13.1.4.243 Unique Identifier (UID)

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume/uid
Type: String

The unique identifier for the volume instance within its deployment context.

© Ecma International 2025

6.13.1.4.244 Name

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume/name
Type: String

The name of the volume instance.

6.13.1.4.245 Mode

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume/mode
Type: String (enum)
Default Value: filesystem

The mode for the volume instance.

Enumeration of possible values:

- filesystem
- block

6.13.1.4.246 Path

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume/path
Type: String

The underlying path created from the actual volume.

6.13.1.4.247 Size allocated

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume/sizeAllocated
Type: String

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.

Example 1 (Informative)

```
10GB
```

Example 2 (Informative)

```
2Ti
```

Example 3 (Informative)

```
1Pi
```

6.13.1.4.248 Persistent

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume/persistent
Type: Boolean

Indicates if the volume persists beyond the life of the resource it is associated with.

© Ecma International 2025

6.13.1.4.249 Remote

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume/remote
Type: Boolean

Indicates if the volume is remotely (i.e., network) attached.

6.13.1.4.250 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.13.1.4.251 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume/properties/[]
Type: Object

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 375 — Properties for the property 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.

6.13.1.4.252 Name

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume/properties/[]/name
Type: String

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

6.13.1.4.253 Value

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/volume/properties/[]/value
Type: String

The value of the property.

© Ecma International 2025

6.13.1.4.254 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.13.1.4.255 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/properties/[]
Type: Object

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 376 — Properties for the property 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.

6.13.1.4.256 Name

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/properties/[]/name
Type: String

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

6.13.1.4.257 Value

Location: /formulation/[]/workflows/[]/tasks/[]/workspaces/[]/properties/[]/value
Type: String

The value of the property.

6.13.1.4.258 Runtime topology

Location: /formulation/[]/workflows/[]/tasks/[]/runtimeTopology
Property: runtimeTopology (Optional)
Type: Array
Uniqueness: All items shall be unique.

A graph of the component runtime topology for task's instance. Each item of this array shall be a Dependency object.

© Ecma International 2025

6.13.1.4.259 Dependency

Location: /formulation/[]/workflows/[]/tasks/[]/runtimeTopology/[]
Type: Object

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 shall 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 377 — 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.

6.13.1.4.260 Reference

Location: /formulation/[]/workflows/[]/tasks/[]/runtimeTopology/[]/ref
Type: String

References a component or service by its bom-ref attribute.

6.13.1.4.261 Depends On

Location: /formulation/[]/workflows/[]/tasks/[]/runtimeTopology/[]/dependsOn
Property: dependsOn (Optional)
Type: Array (of String)
Uniqueness: All items shall be unique.

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 shall be a string.

6.13.1.4.262 Provides

Location: /formulation/[]/workflows/[]/tasks/[]/runtimeTopology/[]/provides
Property: provides (Optional)
Type: Array (of String)
Uniqueness: All items shall be unique.

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 shall be a string.

© Ecma International 2025

6.13.1.4.263 Properties

Location: /formulation/[]/workflows/[]/tasks/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.13.1.4.264 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/tasks/[]/properties/[]
Type: Object

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 property 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.

6.13.1.4.265 Name

Location: /formulation/[]/workflows/[]/tasks/[]/properties/[]/name
Type: String

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

6.13.1.4.266 Value

Location: /formulation/[]/workflows/[]/tasks/[]/properties/[]/value
Type: String

The value of the property.

6.13.1.4.267 Task dependency graph

Location: /formulation/[]/workflows/[]/taskDependencies
Property: taskDependencies (Optional)
Type: Array
Uniqueness: All items shall be unique.

The graph of dependencies between tasks within the workflow. Each item of this array shall be a Dependency object.

© Ecma International 2025

6.13.1.4.268 Dependency

Location: /formulation/[]/workflows/[]/taskDependencies/[]
Type: Object

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 shall 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 379 — Properties for the taskDependency 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.

6.13.1.4.269 Reference

Location: /formulation/[]/workflows/[]/taskDependencies/[]/ref
Type: String

References a component or service by its bom-ref attribute.

6.13.1.4.270 Depends On

Location: /formulation/[]/workflows/[]/taskDependencies/[]/dependsOn
Property: dependsOn (Optional)
Type: Array (of String)
Uniqueness: All items shall be unique.

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 shall be a string.

6.13.1.4.271 Provides

Location: /formulation/[]/workflows/[]/taskDependencies/[]/provides
Property: provides (Optional)
Type: Array (of String)
Uniqueness: All items shall be unique.

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 shall be a string.

© Ecma International 2025

6.13.1.4.272 Task types

Location: /formulation/[]/workflows/[]/taskTypes
Property: taskTypes (Required)
Type: Array (of String) (enum)

Indicates the types of activities performed by the set of workflow tasks. Each item of this array shall be a string.

Table 380 — 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.

6.13.1.4.273 Trigger

Location: /formulation/[]/workflows/[]/trigger
Type: Object

The trigger that initiated the task.

Table 381 — Properties for the trigger object

Property	Type	Requirement	Description
bom-ref	String	Required	An identifier which can be used to reference the trigger elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. 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.

© Ecma International 2025

Table 381 — Properties for the trigger object *(continued)*

Property	Type	Requirement	Description
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.

6.13.1.4.274 BOM Reference

Location: /formulation/[]/workflows/[]/trigger/bom-ref
Type: String

An identifier which can be used to reference the trigger elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. Value should not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

6.13.1.4.275 Unique Identifier (UID)

Location: /formulation/[]/workflows/[]/trigger/uid
Type: String

The unique identifier for the resource instance within its deployment context.

6.13.1.4.276 Name

Location: /formulation/[]/workflows/[]/trigger/name
Type: String

The name of the resource instance.

6.13.1.4.277 Description

Location: /formulation/[]/workflows/[]/trigger/description
Type: String

A description of the resource instance.

© Ecma International 2025

6.13.1.4.278 Resource references

Location: /formulation/[]/workflows/[]/trigger/resourceReferences
Property: resourceReferences (Optional)
Type: Array
Uniqueness: All items shall be unique.

References to component or service resources that are used to realize the resource instance. Each item of this array shall be a Resource reference choice object.

6.13.1.4.279 Resource reference choice

Location: /formulation/[]/workflows/[]/trigger/resourceReferences/[]
Type: Object

A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Table 382 — Properties for the resourceReference object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.280 BOM Reference

Location: /formulation/[]/workflows/[]/trigger/resourceReferences/[]/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.281 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.282 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.283 External reference

Location: /formulation/[]/workflows/[]/trigger/resourceReferences/[]/externalReference
Type: Object

Reference to an externally accessible resource.

© Ecma International 2025

6.13.1.4.284 Type

Location: /formulation/[]/workflows/[]/trigger/type
Type: String (enum)

The source type of event which caused the trigger to fire.

Enumeration of possible values:

- manual
- api
- webhook
- scheduled

6.13.1.4.285 Event

Location: /formulation/[]/workflows/[]/trigger/event
Type: Object

The event data that caused the associated trigger to activate.

Table 383 — 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	Encoding of the raw event data.
source	Array	Optional	References the component or service that was the source of the event.
target	Array	Optional	References the component or service that was the target of the event.
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.

6.13.1.4.286 Unique Identifier (UID)

Location: /formulation/[]/workflows/[]/trigger/event/uid
Type: String

The unique identifier of the event.

6.13.1.4.287 Description

Location: /formulation/[]/workflows/[]/trigger/event/description
Type: String

A description of the event.

© Ecma International 2025

6.13.1.4.288 Time Received

Location: /formulation/[]/workflows/[]/trigger/event/timeReceived
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the event was received.

6.13.1.4.289 Data

Location: /formulation/[]/workflows/[]/trigger/event/data
Type: Object

Encoding of the raw event data.

Table 384 — 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 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.

6.13.1.4.290 Content-Type

Location: /formulation/[]/workflows/[]/trigger/event/data/contentType
Type: String
Default Value: text/plain

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.

Example 1 (Informative)

```
text/plain
```

Example 2 (Informative)

```
application/json
```

Example 3 (Informative)

```
image/png
```

6.13.1.4.291 Encoding

Location: /formulation/[]/workflows/[]/trigger/event/data/encoding
Type: String (enum)

Specifies the encoding the text is represented in.

© Ecma International 2025

Table 385 — Enumeration of possible values

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

6.13.1.4.292 Attachment Text

Location: /formulation/[]/workflows/[]/trigger/event/data/content
Type: String

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

6.13.1.4.293 Source

Location: /formulation/[]/workflows/[]/trigger/event/source
Type: Object

References the component or service that was the source of the event.

Table 386 — Properties for the source object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.294 BOM Reference

Location: /formulation/[]/workflows/[]/trigger/event/source/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.295 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.296 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

© Ecma International 2025

6.13.1.4.297 External reference

Location: /formulation/[]/workflows/[]/trigger/event/source/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.298 Target

Location: /formulation/[]/workflows/[]/trigger/event/target
Type: Object

References the component or service that was the target of the event.

Table 387 — Properties for the target object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.299 BOM Reference

Location: /formulation/[]/workflows/[]/trigger/event/target/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.300 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.301 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.302 External reference

Location: /formulation/[]/workflows/[]/trigger/event/target/externalReference
Type: Object

Reference to an externally accessible resource.

© Ecma International 2025

6.13.1.4.303 Properties

Location: /formulation/[]/workflows/[]/trigger/event/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.13.1.4.304 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/trigger/event/properties/[]
Type: Object

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 388 — Properties for the property 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.

6.13.1.4.305 Name

Location: /formulation/[]/workflows/[]/trigger/event/properties/[]/name
Type: String

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

6.13.1.4.306 Value

Location: /formulation/[]/workflows/[]/trigger/event/properties/[]/value
Type: String

The value of the property.

6.13.1.4.307 Conditions

Location: /formulation/[]/workflows/[]/trigger/conditions
Property: conditions (Optional)
Type: Array
Uniqueness: All items shall be unique.

A list of conditions used to determine if a trigger should be activated. Each item of this array shall be a Condition object.

© Ecma International 2025

6.13.1.4.308 Condition

Location: /formulation/[]/workflows/[]/trigger/conditions/[]
Type: Object

A condition that was used to determine a trigger should be activated.

Table 389 — Properties for the condition 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.

6.13.1.4.309 Description

Location: /formulation/[]/workflows/[]/trigger/conditions/[]/description
Type: String

Describes the set of conditions which cause the trigger to activate.

6.13.1.4.310 Expression

Location: /formulation/[]/workflows/[]/trigger/conditions/[]/expression
Type: String

The logical expression that was evaluated that determined the trigger should be fired.

6.13.1.4.311 Properties

Location: /formulation/[]/workflows/[]/trigger/conditions/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.13.1.4.312 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/trigger/conditions/[]/properties/[]
Type: Object

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

© Ecma International 2025

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

Table 390 — Properties for the property 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.

6.13.1.4.313 Name

Location: /formulation/[]/workflows/[]/trigger/conditions/[]/properties/[]/name
Type: String

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

6.13.1.4.314 Value

Location: /formulation/[]/workflows/[]/trigger/conditions/[]/properties/[]/value
Type: String

The value of the property.

6.13.1.4.315 Time activated

Location: /formulation/[]/workflows/[]/trigger/timeActivated
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the trigger was activated.

6.13.1.4.316 Inputs

Location: /formulation/[]/workflows/[]/trigger/inputs
Property: inputs (Optional)
Type: Array
Uniqueness: All items shall be unique.

Represents resources and data brought into a task at runtime by executor or task commands Each item of this array shall be an Input type object.

6.13.1.4.317 Input type

Location: /formulation/[]/workflows/[]/trigger/inputs/[]
Type: Object

Type that represents various input data types and formats.

© Ecma International 2025

Table 391 — Properties for the input object

Property	Type	Requirement	Description
source	Array	Optional	A reference to the component or service that provided the input to the task (e.g., reference to a service with data flow value of **`inbound`**).
target	Array	Optional	A reference to the component or service that received or stored the input if not the task itself (e.g., a local, named storage workspace).
resource	Array	Optional	A reference to an independent resource provided as an input to a task by the workflow runtime.
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	Inputs that have the form of data.
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.

6.13.1.4.318 Source

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/source
Type: Object

A reference to the component or service that provided the input to the task (e.g., reference to a service with data flow value of **`inbound`**).

Example 1 (Informative)

```
source code repository
```

Example 2 (Informative)

```
database
```

Table 392 — Properties for the source object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.319 BOM Reference

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/source/ref

References an object by its bom-ref attribute.

© Ecma International 2025

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.320 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.321 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.322 External reference

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/source/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.323 Target

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/target
Type: Object

A reference to the component or service that received or stored the input if not the task itself (e.g., a local, named storage workspace).

Example 1 (Informative)
`workspace`

Example 2 (Informative)
`directory`

Table 393 — Properties for the target object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.324 BOM Reference

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/target/ref

References an object by its bom-ref attribute.

© Ecma International 2025

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.325 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.326 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.327 External reference

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/target/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.328 Resource

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/resource
Type: Object

A reference to an independent resource provided as an input to a task by the workflow runtime.

Example 1 (Informative)
`a reference to a configuration file in a repository (i.e., a bom-ref)`

Example 2 (Informative)
`a reference to a scanning service used in a task (i.e., a bom-ref)`

Table 394 — Properties for the resource object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.329 BOM Reference

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/resource/ref

References an object by its bom-ref attribute.

© Ecma International 2025

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.330 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.331 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.332 External reference

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/resource/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.333 Parameters

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/parameters
Property: parameters (Optional)
Type: Array
Uniqueness: All items shall be unique.

Inputs that have the form of parameters with names and values. Each item of this array shall be a Parameter object.

6.13.1.4.334 Parameter

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/parameters/[]
Type: Object

A representation of a functional parameter.

Table 395 — Properties for the parameter 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.

© Ecma International 2025

6.13.1.4.335 Name

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/parameters/[]/name
Type: String

The name of the parameter.

6.13.1.4.336 Value

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/parameters/[]/value
Type: String

The value of the parameter.

6.13.1.4.337 Data type

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/parameters/[]/dataType
Type: String

The data type of the parameter.

6.13.1.4.338 Environment variables

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/environmentVars
Property: environmentVars (Optional)
Type: Array
Uniqueness: All items shall be unique.

Inputs that have the form of parameters with names and values.

6.13.1.4.339 EnvironmentVar

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/environmentVars/[]

Shall be one of:

1. Lightweight name-value pair
2. String-Based Environment Variables

6.13.1.4.340 Lightweight name-value pair

Type: Object

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 396 — Properties for the lightweight name-value pair 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.

© Ecma International 2025

6.13.1.4.341 String-Based Environment Variables

Type: String

In addition to the more common key–value pair format, some environment variables may consist of a single string without an explicit value assignment. These string-based environment variables typically act as flags or signals to software, indicating that a feature should be enabled, a mode should be activated, or a specific condition is present. Their presence alone conveys meaning.

6.13.1.4.342 Name

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/environmentVars/[]/name
Property: name (Required)
Type: String

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

6.13.1.4.343 Value

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/environmentVars/[]/value
Property: value (Optional)
Type: String

The value of the property.

6.13.1.4.344 Data

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/data
Type: Object

Inputs that have the form of data.

Table 397 — 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 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.

6.13.1.4.345 Content-Type

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/data/contentType
Type: String
Default Value: text/plain

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.

© Ecma International 2025

Example 1 (Informative)
`text/plain`

Example 2 (Informative)
`application/json`

Example 3 (Informative)
`image/png`

6.13.1.4.346 Encoding

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/data/encoding
Type: String (enum)

Specifies the encoding the text is represented in.

Table 398 — Enumeration of possible values

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

6.13.1.4.347 Attachment Text

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/data/content
Type: String

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

6.13.1.4.348 Properties

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.13.1.4.349 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/properties/[]
Type: Object

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.

© Ecma International 2025

Table 399 — Properties for the property 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.

6.13.1.4.350 Name

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/properties/[]/name
Type: String

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

6.13.1.4.351 Value

Location: /formulation/[]/workflows/[]/trigger/inputs/[]/properties/[]/value
Type: String

The value of the property.

6.13.1.4.352 Outputs

Location: /formulation/[]/workflows/[]/trigger/outputs
Property: outputs (Optional)
Type: Array
Uniqueness: All items shall be unique.

Represents resources and data output from a task at runtime by executor or task commands

6.13.1.4.353 Output

Location: /formulation/[]/workflows/[]/trigger/outputs/[]
Type: Object

Table 400 — Properties for the output object

Property	Type	Requirement	Description
type	String	Optional	Describes the type of data output.
source	Array	Optional	Component or service that generated or provided the output from the task (e.g., a build tool).
target	Array	Optional	Component or service that received the output from the task (e.g., reference to an artifactory service with data flow value of **`outbound`**).
resource	Array	Optional	A reference to an independent resource generated as output by the task.
data	Object	Optional	Outputs that have the form of data.

© Ecma International 2025

Table 400 — Properties for the output object *(continued)*

Property	Type	Requirement	Description
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.

6.13.1.4.354 Type

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/type
Type: String (enum)

Describes the type of data output.

Enumeration of possible values:

- artifact
- attestation
- log
- evidence
- metrics
- other

6.13.1.4.355 Source

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/source
Type: Object

Component or service that generated or provided the output from the task (e.g., a build tool).

Table 401 — Properties for the source object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.356 BOM Reference

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/source/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

© Ecma International 2025

6.13.1.4.357 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.358 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.359 External reference

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/source/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.360 Target

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/target
Type: Object

Component or service that received the output from the task (e.g., reference to an artifactory service with data flow value of **`outbound`**).

Example (Informative)

```
a log file described as an `externalReference` within its target domain.
```

Table 402 — Properties for the target object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.361 BOM Reference

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/target/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

© Ecma International 2025

6.13.1.4.362 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.363 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.364 External reference

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/target/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.365 Resource

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/resource
Type: Object

A reference to an independent resource generated as output by the task.

Example 1 (Informative)
`configuration file`

Example 2 (Informative)
`source code`

Example 3 (Informative)
`scanning service`

Table 403 — Properties for the resource object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.366 BOM Reference

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/resource/ref

References an object by its bom-ref attribute.

© Ecma International 2025

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.367 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.368 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.369 External reference

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/resource/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.370 Data

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/data
Type: Object

Outputs that have the form of data.

Table 404 — 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 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.

© Ecma International 2025

6.13.1.4.371 Content-Type

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/data/contentType
Type: String
Default Value: text/plain

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.

Example 1 (Informative)
```
text/plain
```

Example 2 (Informative)
```
application/json
```

Example 3 (Informative)
```
image/png
```

6.13.1.4.372 Encoding

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/data/encoding
Type: String (enum)

Specifies the encoding the text is represented in.

Table 405 — Enumeration of possible values

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

6.13.1.4.373 Attachment Text

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/data/content
Type: String

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

6.13.1.4.374 Environment variables

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/environmentVars
Property: environmentVars (Optional)
Type: Array
Uniqueness: All items shall be unique.

Outputs that have the form of environment variables.

6.13.1.4.375 EnvironmentVar

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/environmentVars/[]

Shall be one of:

© Ecma International 2025

1. Lightweight name-value pair
2. String-Based Environment Variables

6.13.1.4.376 Lightweight name-value pair

Type: Object

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 406 — Properties for the lightweight name-value pair 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.

6.13.1.4.377 String-Based Environment Variables

Type: String

In addition to the more common key–value pair format, some environment variables may consist of a single string without an explicit value assignment. These string-based environment variables typically act as flags or signals to software, indicating that a feature should be enabled, a mode should be activated, or a specific condition is present. Their presence alone conveys meaning.

6.13.1.4.378 Name

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/environmentVars/[]/name
Property: name (Required)
Type: String

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

6.13.1.4.379 Value

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/environmentVars/[]/value
Property: value (Optional)
Type: String

The value of the property.

6.13.1.4.380 Properties

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

© Ecma International 2025

6.13.1.4.381 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/properties/[]
Type: Object

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 407 — Properties for the property 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.

6.13.1.4.382 Name

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/properties/[]/name
Type: String

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

6.13.1.4.383 Value

Location: /formulation/[]/workflows/[]/trigger/outputs/[]/properties/[]/value
Type: String

The value of the property.

6.13.1.4.384 Properties

Location: /formulation/[]/workflows/[]/trigger/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.13.1.4.385 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/trigger/properties/[]
Type: Object

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.

© Ecma International 2025

Table 408 — Properties for the property 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.

6.13.1.4.386 Name

Location: /formulation/[]/workflows/[]/trigger/properties/[]/name
Type: String

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

6.13.1.4.387 Value

Location: /formulation/[]/workflows/[]/trigger/properties/[]/value
Type: String

The value of the property.

6.13.1.4.388 Steps

Location: /formulation/[]/workflows/[]/steps
Property: steps (Optional)
Type: Array
Uniqueness: All items shall be unique.

The sequence of steps for the task.

6.13.1.4.389 Step

Location: /formulation/[]/workflows/[]/steps/[]
Type: Object

Executes specific commands or tools in order to accomplish its owning task as part of a sequence.

Table 409 — Properties for the step 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.
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.

© Ecma International 2025

6.13.1.4.390 Name

Location: /formulation/[]/workflows/[]/steps/[]/name
Type: String

A name for the step.

6.13.1.4.391 Description

Location: /formulation/[]/workflows/[]/steps/[]/description
Type: String

A description of the step.

6.13.1.4.392 Commands

Location: /formulation/[]/workflows/[]/steps/[]/commands
Property: commands (Optional)
Type: Array

Ordered list of commands or directives for the step

6.13.1.4.393 Command

Location: /formulation/[]/workflows/[]/steps/[]/commands/[]
Type: Object

Table 410 — Properties for the command 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.

6.13.1.4.394 Executed

Location: /formulation/[]/workflows/[]/steps/[]/commands/[]/executed
Type: String

A text representation of the executed command.

6.13.1.4.395 Properties

Location: /formulation/[]/workflows/[]/steps/[]/commands/[]/properties
Property: properties (Optional)
Type: Array

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

© Ecma International 2025

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 shall be a Lightweight name-value pair object.

6.13.1.4.396 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/steps/[]/commands/[]/properties/[]
Type: Object

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 411 — Properties for the property 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.

6.13.1.4.397 Name

Location: /formulation/[]/workflows/[]/steps/[]/commands/[]/properties/[]/name
Type: String

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

6.13.1.4.398 Value

Location: /formulation/[]/workflows/[]/steps/[]/commands/[]/properties/[]/value
Type: String

The value of the property.

6.13.1.4.399 Properties

Location: /formulation/[]/workflows/[]/steps/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.13.1.4.400 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/steps/[]/properties/[]
Type: Object

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

© Ecma International 2025

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

Table 412 — Properties for the property 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.

6.13.1.4.401 Name

Location: /formulation/[]/workflows/[]/steps/[]/properties/[]/name
Type: String

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

6.13.1.4.402 Value

Location: /formulation/[]/workflows/[]/steps/[]/properties/[]/value
Type: String

The value of the property.

6.13.1.4.403 Inputs

Location: /formulation/[]/workflows/[]/inputs
Property: inputs (Optional)
Type: Array
Uniqueness: All items shall be unique.

Represents resources and data brought into a task at runtime by executor or task commands Each item of this array shall be an Input type object.

6.13.1.4.404 Input type

Location: /formulation/[]/workflows/[]/inputs/[]
Type: Object

Type that represents various input data types and formats.

Table 413 — Properties for the input object

Property	Type	Requirement	Description
source	Array	Optional	A reference to the component or service that provided the input to the task (e.g., reference to a service with data flow value of **`inbound`**).
target	Array	Optional	A reference to the component or service that received or stored the input if not the task itself (e.g., a local, named storage workspace).
resource	Array	Optional	A reference to an independent resource provided as an input to a task by the workflow runtime.
parameters	Array	Optional	Inputs that have the form of parameters with names and values.

© Ecma International 2025

Table 413 — Properties for the input object *(continued)*

Property	Type	Requirement	Description
environmentVars	Array	Optional	Inputs that have the form of parameters with names and values.
data	Object	Optional	Inputs that have the form of data.
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.

6.13.1.4.405 Source

Location: /formulation/[]/workflows/[]/inputs/[]/source
Type: Object

A reference to the component or service that provided the input to the task (e.g., reference to a service with data flow value of **`inbound`**).

Example 1 (Informative)

```
source code repository
```

Example 2 (Informative)

```
database
```

Table 414 — Properties for the source object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.406 BOM Reference

Location: /formulation/[]/workflows/[]/inputs/[]/source/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.407 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

© Ecma International 2025

6.13.1.4.408 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.409 External reference

Location: /formulation/[]/workflows/[]/inputs/[]/source/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.410 Target

Location: /formulation/[]/workflows/[]/inputs/[]/target
Type: Object

A reference to the component or service that received or stored the input if not the task itself (e.g., a local, named storage workspace).

Example 1 (Informative)

```
workspace
```

Example 2 (Informative)

```
directory
```

Table 415 — Properties for the target object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.411 BOM Reference

Location: /formulation/[]/workflows/[]/inputs/[]/target/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.412 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

© Ecma International 2025

6.13.1.4.413 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.414 External reference

Location: /formulation/[]/workflows/[]/inputs/[]/target/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.415 Resource

Location: /formulation/[]/workflows/[]/inputs/[]/resource
Type: Object

A reference to an independent resource provided as an input to a task by the workflow runtime.

Example 1 (Informative)

```
a reference to a configuration file in a repository (i.e., a bom-ref)
```

Example 2 (Informative)

```
a reference to a scanning service used in a task (i.e., a bom-ref)
```

Table 416 — Properties for the resource object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.416 BOM Reference

Location: /formulation/[]/workflows/[]/inputs/[]/resource/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.417 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

© Ecma International 2025

6.13.1.4.418 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.419 External reference

Location: /formulation/[]/workflows/[]/inputs/[]/resource/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.420 Parameters

Location: /formulation/[]/workflows/[]/inputs/[]/parameters
Property: parameters (Optional)
Type: Array
Uniqueness: All items shall be unique.

Inputs that have the form of parameters with names and values. Each item of this array shall be a Parameter object.

6.13.1.4.421 Parameter

Location: /formulation/[]/workflows/[]/inputs/[]/parameters/[]
Type: Object

A representation of a functional parameter.

Table 417 — Properties for the parameter 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.

6.13.1.4.422 Name

Location: /formulation/[]/workflows/[]/inputs/[]/parameters/[]/name
Type: String

The name of the parameter.

6.13.1.4.423 Value

Location: /formulation/[]/workflows/[]/inputs/[]/parameters/[]/value
Type: String

The value of the parameter.

© Ecma International 2025

6.13.1.4.424 Data type

Location: /formulation/[]/workflows/[]/inputs/[]/parameters/[]/dataType
Type: String

The data type of the parameter.

6.13.1.4.425 Environment variables

Location: /formulation/[]/workflows/[]/inputs/[]/environmentVars
Property: environmentVars (Optional)
Type: Array
Uniqueness: All items shall be unique.

Inputs that have the form of parameters with names and values.

6.13.1.4.426 EnvironmentVar

Location: /formulation/[]/workflows/[]/inputs/[]/environmentVars/[]

Shall be one of:

1. Lightweight name-value pair
2. String-Based Environment Variables

6.13.1.4.427 Lightweight name-value pair

Type: Object

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 418 — Properties for the lightweight name-value pair 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.

6.13.1.4.428 String-Based Environment Variables

Type: String

In addition to the more common key–value pair format, some environment variables may consist of a single string without an explicit value assignment. These string-based environment variables typically act as flags or signals to software, indicating that a feature should be enabled, a mode should be activated, or a specific condition is present. Their presence alone conveys meaning.

© Ecma International 2025

6.13.1.4.429 Name

Location: /formulation/[]/workflows/[]/inputs/[]/environmentVars/[]/name
Property: name (Required)
Type: String

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

6.13.1.4.430 Value

Location: /formulation/[]/workflows/[]/inputs/[]/environmentVars/[]/value
Property: value (Optional)
Type: String

The value of the property.

6.13.1.4.431 Data

Location: /formulation/[]/workflows/[]/inputs/[]/data
Type: Object

Inputs that have the form of data.

Table 419 — 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 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.

6.13.1.4.432 Content-Type

Location: /formulation/[]/workflows/[]/inputs/[]/data/contentType
Type: String
Default Value: text/plain

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.

Example 1 (Informative)

```
text/plain
```

Example 2 (Informative)

```
application/json
```

© Ecma International 2025

Example 3 (Informative)
`image/png`

6.13.1.4.433 Encoding

Location: /formulation/[]/workflows/[]/inputs/[]/data/encoding
Type: String (enum)

Specifies the encoding the text is represented in.

Table 420 — Enumeration of possible values

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

6.13.1.4.434 Attachment Text

Location: /formulation/[]/workflows/[]/inputs/[]/data/content
Type: String

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

6.13.1.4.435 Properties

Location: /formulation/[]/workflows/[]/inputs/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.13.1.4.436 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/inputs/[]/properties/[]
Type: Object

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 421 — Properties for the property 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.

© Ecma International 2025

6.13.1.4.437 Name

Location: /formulation/[]/workflows/[]/inputs/[]/properties/[]/name
Type: String

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

6.13.1.4.438 Value

Location: /formulation/[]/workflows/[]/inputs/[]/properties/[]/value
Type: String

The value of the property.

6.13.1.4.439 Outputs

Location: /formulation/[]/workflows/[]/outputs
Property: outputs (Optional)
Type: Array
Uniqueness: All items shall be unique.

Represents resources and data output from a task at runtime by executor or task commands

6.13.1.4.440 Output

Location: /formulation/[]/workflows/[]/outputs/[]
Type: Object

Table 422 — Properties for the output object

Property	Type	Requirement	Description
type	String	Optional	Describes the type of data output.
source	Array	Optional	Component or service that generated or provided the output from the task (e.g., a build tool).
target	Array	Optional	Component or service that received the output from the task (e.g., reference to an artifactory service with data flow value of **`outbound`**).
resource	Array	Optional	A reference to an independent resource generated as output by the task.
data	Object	Optional	Outputs that have the form of data.
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.

© Ecma International 2025

6.13.1.4.441 Type

Location: /formulation/[]/workflows/[]/outputs/[]/type
Type: String (enum)

Describes the type of data output.

Enumeration of possible values:

- artifact
- attestation
- log
- evidence
- metrics
- other

6.13.1.4.442 Source

Location: /formulation/[]/workflows/[]/outputs/[]/source
Type: Object

Component or service that generated or provided the output from the task (e.g., a build tool).

Table 423 — Properties for the source object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.443 BOM Reference

Location: /formulation/[]/workflows/[]/outputs/[]/source/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.444 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.445 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

© Ecma International 2025

6.13.1.4.446 External reference

Location: /formulation/[]/workflows/[]/outputs/[]/source/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.447 Target

Location: /formulation/[]/workflows/[]/outputs/[]/target
Type: Object

Component or service that received the output from the task (e.g., reference to an artifactory service with data flow value of **`outbound`**).

Example (Informative)

```
a log file described as an `externalReference` within its target domain.
```

Table 424 — Properties for the target object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.448 BOM Reference

Location: /formulation/[]/workflows/[]/outputs/[]/target/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.449 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.450 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

© Ecma International 2025

6.13.1.4.451 External reference

Location: /formulation/[]/workflows/[]/outputs/[]/target/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.452 Resource

Location: /formulation/[]/workflows/[]/outputs/[]/resource
Type: Object

A reference to an independent resource generated as output by the task.

Example 1 (Informative)
```
configuration file
```

Example 2 (Informative)
```
source code
```

Example 3 (Informative)
```
scanning service
```

Table 425 — Properties for the resource object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.453 BOM Reference

Location: /formulation/[]/workflows/[]/outputs/[]/resource/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

6.13.1.4.454 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

© Ecma International 2025

6.13.1.4.455 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.456 External reference

Location: /formulation/[]/workflows/[]/outputs/[]/resource/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.457 Data

Location: /formulation/[]/workflows/[]/outputs/[]/data
Type: Object

Outputs that have the form of data.

Table 426 — 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 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.

6.13.1.4.458 Content-Type

Location: /formulation/[]/workflows/[]/outputs/[]/data/contentType
Type: String
Default Value: text/plain

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.

Example 1 (Informative)

```
text/plain
```

Example 2 (Informative)

```
application/json
```

© Ecma International 2025

Example 3 (Informative)
`image/png`

6.13.1.4.459 Encoding

Location: /formulation/[]/workflows/[]/outputs/[]/data/encoding
Type: String (enum)

Specifies the encoding the text is represented in.

Table 427 — Enumeration of possible values

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

6.13.1.4.460 Attachment Text

Location: /formulation/[]/workflows/[]/outputs/[]/data/content
Type: String

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

6.13.1.4.461 Environment variables

Location: /formulation/[]/workflows/[]/outputs/[]/environmentVars
Property: environmentVars (Optional)
Type: Array
Uniqueness: All items shall be unique.

Outputs that have the form of environment variables.

6.13.1.4.462 EnvironmentVar

Location: /formulation/[]/workflows/[]/outputs/[]/environmentVars/[]

Shall be one of:

1. Lightweight name-value pair
2. String-Based Environment Variables

6.13.1.4.463 Lightweight name-value pair

Type: Object

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.

© Ecma International 2025

Table 428 — Properties for the lightweight name-value pair 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.

6.13.1.4.464 String-Based Environment Variables

Type: String

In addition to the more common key–value pair format, some environment variables may consist of a single string without an explicit value assignment. These string-based environment variables typically act as flags or signals to software, indicating that a feature should be enabled, a mode should be activated, or a specific condition is present. Their presence alone conveys meaning.

6.13.1.4.465 Name

Location: /formulation/[]/workflows/[]/outputs/[]/environmentVars/[]/name
Property: name (Required)
Type: String

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

6.13.1.4.466 Value

Location: /formulation/[]/workflows/[]/outputs/[]/environmentVars/[]/value
Property: value (Optional)
Type: String

The value of the property.

6.13.1.4.467 Properties

Location: /formulation/[]/workflows/[]/outputs/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.13.1.4.468 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/outputs/[]/properties/[]
Type: Object

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.

© Ecma International 2025

Table 429 — Properties for the property 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.

6.13.1.4.469 Name

Location: /formulation/[]/workflows/[]/outputs/[]/properties/[]/name
Type: String

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

6.13.1.4.470 Value

Location: /formulation/[]/workflows/[]/outputs/[]/properties/[]/value
Type: String

The value of the property.

6.13.1.4.471 Time start

Location: /formulation/[]/workflows/[]/timeStart
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the task started.

6.13.1.4.472 Time end

Location: /formulation/[]/workflows/[]/timeEnd
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the task ended.

6.13.1.4.473 Workspaces

Location: /formulation/[]/workflows/[]/workspaces
Property: workspaces (Optional)
Type: Array
Uniqueness: All items shall be unique.

A set of named filesystem or data resource shareable by workflow tasks. Each item of this array shall be a Workspace object.

6.13.1.4.474 Workspace

Location: /formulation/[]/workflows/[]/workspaces/[]
Type: Object

A named filesystem or data resource shareable by workflow tasks.

© Ecma International 2025

Table 430 — Properties for the workspace object

Property	Type	Requirement	Description
bom-ref	String	Required	An identifier which can be used to reference the workspace elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. 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.

6.13.1.4.475 BOM Reference

Location: /formulation/[]/workflows/[]/workspaces/[]/bom-ref
Type: String

An identifier which can be used to reference the workspace elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. Value should not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

6.13.1.4.476 Unique Identifier (UID)

Location: /formulation/[]/workflows/[]/workspaces/[]/uid
Type: String

The unique identifier for the resource instance within its deployment context.

© Ecma International 2025

6.13.1.4.477 Name

Location: /formulation/[]/workflows/[]/workspaces/[]/name
Type: String

The name of the resource instance.

6.13.1.4.478 Aliases

Location: /formulation/[]/workflows/[]/workspaces/[]/aliases
Property: aliases (Optional)
Type: Array (of String)

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 shall be a string.

6.13.1.4.479 Description

Location: /formulation/[]/workflows/[]/workspaces/[]/description
Type: String

A description of the resource instance.

6.13.1.4.480 Resource references

Location: /formulation/[]/workflows/[]/workspaces/[]/resourceReferences
Property: resourceReferences (Optional)
Type: Array
Uniqueness: All items shall be unique.

References to component or service resources that are used to realize the resource instance. Each item of this array shall be a Resource reference choice object.

6.13.1.4.481 Resource reference choice

Location: /formulation/[]/workflows/[]/workspaces/[]/resourceReferences/[]
Type: Object

A reference to a locally defined resource (e.g., a bom-ref) or an externally accessible resource.

Table 431 — Properties for the resourceReference object

Property	Type	Requirement	Description
ref	Array	Optional	References an object by its bom-ref attribute.
externalReference	Object	Optional	Reference to an externally accessible resource.

6.13.1.4.482 BOM Reference

Location: /formulation/[]/workflows/[]/workspaces/[]/resourceReferences/[]/ref

References an object by its bom-ref attribute.

Shall be any of:

1. Ref
2. BOM-Link Element

© Ecma International 2025

6.13.1.4.483 Ref

Type: String

Descriptor for an element identified by the attribute 'bom-ref' in the same BOM document. In contrast to **`bomLinkElementType`**.

6.13.1.4.484 BOM-Link Element

Type: String
Format: iri-reference as specified in RFC 3987
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]*#.+`

Descriptor for an element in a BOM document. See https://cyclonedx.org/capabilities/bomlink/.

6.13.1.4.485 External reference

Location: /formulation/[]/workflows/[]/workspaces/[]/resourceReferences/[]/externalReference
Type: Object

Reference to an externally accessible resource.

6.13.1.4.486 Access mode

Location: /formulation/[]/workflows/[]/workspaces/[]/accessMode
Type: String (enum)

Describes the read-write access control for the workspace relative to the owning resource instance.

Enumeration of possible values:

- read-only
- read-write
- read-write-once
- write-once
- write-only

6.13.1.4.487 Mount path

Location: /formulation/[]/workflows/[]/workspaces/[]/mountPath
Type: String

A path to a location on disk where the workspace will be available to the associated task's steps.

6.13.1.4.488 Managed data type

Location: /formulation/[]/workflows/[]/workspaces/[]/managedDataType
Type: String

The name of a domain-specific data type the workspace represents.

Example 1 (Informative)
`ConfigMap`

© Ecma International 2025

Example 2 (Informative)

```
Secret
```

6.13.1.4.489 Volume request

Location: /formulation/[]/workflows/[]/workspaces/[]/volumeRequest
Type: String

Identifies the reference to the request for a specific volume type and parameters.

Example (Informative)

```
a kubernetes Persistent Volume Claim (PVC) name
```

6.13.1.4.490 Volume

Location: /formulation/[]/workflows/[]/workspaces/[]/volume
Type: Object

Information about the actual volume instance allocated to the workspace.

Example (Informative)

```
see https://kubernetes.io/docs/concepts/storage/persistent-volumes/
```

Table 432 — 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.

© Ecma International 2025

6.13.1.4.491 Unique Identifier (UID)

Location: /formulation/[]/workflows/[]/workspaces/[]/volume/uid
Type: String

The unique identifier for the volume instance within its deployment context.

6.13.1.4.492 Name

Location: /formulation/[]/workflows/[]/workspaces/[]/volume/name
Type: String

The name of the volume instance.

6.13.1.4.493 Mode

Location: /formulation/[]/workflows/[]/workspaces/[]/volume/mode
Type: String (enum)
Default Value: filesystem

The mode for the volume instance.

Enumeration of possible values:

- filesystem
- block

6.13.1.4.494 Path

Location: /formulation/[]/workflows/[]/workspaces/[]/volume/path
Type: String

The underlying path created from the actual volume.

6.13.1.4.495 Size allocated

Location: /formulation/[]/workflows/[]/workspaces/[]/volume/sizeAllocated
Type: String

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.

Example 1 (Informative)

```
10GB
```

Example 2 (Informative)

```
2Ti
```

Example 3 (Informative)

```
1Pi
```

© Ecma International 2025

6.13.1.4.496 Persistent

Location: /formulation/[]/workflows/[]/workspaces/[]/volume/persistent
Type: Boolean

Indicates if the volume persists beyond the life of the resource it is associated with.

6.13.1.4.497 Remote

Location: /formulation/[]/workflows/[]/workspaces/[]/volume/remote
Type: Boolean

Indicates if the volume is remotely (i.e., network) attached.

6.13.1.4.498 Properties

Location: /formulation/[]/workflows/[]/workspaces/[]/volume/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.13.1.4.499 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/workspaces/[]/volume/properties/[]
Type: Object

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 433 — Properties for the property 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.

6.13.1.4.500 Name

Location: /formulation/[]/workflows/[]/workspaces/[]/volume/properties/[]/name
Type: String

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

© Ecma International 2025

6.13.1.4.501 Value

Location: /formulation/[]/workflows/[]/workspaces/[]/volume/properties/[]/value
Type: String

The value of the property.

6.13.1.4.502 Properties

Location: /formulation/[]/workflows/[]/workspaces/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.13.1.4.503 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/workspaces/[]/properties/[]
Type: Object

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 434 — Properties for the property 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.

6.13.1.4.504 Name

Location: /formulation/[]/workflows/[]/workspaces/[]/properties/[]/name
Type: String

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

6.13.1.4.505 Value

Location: /formulation/[]/workflows/[]/workspaces/[]/properties/[]/value
Type: String

The value of the property.

© Ecma International 2025

6.13.1.4.506 Runtime topology

Location: /formulation/[]/workflows/[]/runtimeTopology
Property: runtimeTopology (Optional)
Type: Array
Uniqueness: All items shall be unique.

A graph of the component runtime topology for workflow's instance. Each item of this array shall be a Dependency object.

6.13.1.4.507 Dependency

Location: /formulation/[]/workflows/[]/runtimeTopology/[]
Type: Object

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 shall 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 435 — 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.

6.13.1.4.508 Reference

Location: /formulation/[]/workflows/[]/runtimeTopology/[]/ref
Type: String

References a component or service by its bom-ref attribute.

6.13.1.4.509 Depends On

Location: /formulation/[]/workflows/[]/runtimeTopology/[]/dependsOn
Property: dependsOn (Optional)
Type: Array (of String)
Uniqueness: All items shall be unique.

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 shall be a string.

© Ecma International 2025

6.13.1.4.510 Provides

Location: /formulation/[]/workflows/[]/runtimeTopology/[]/provides
Property: provides (Optional)
Type: Array (of String)
Uniqueness: All items shall be unique.

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 shall be a string.

6.13.1.4.511 Properties

Location: /formulation/[]/workflows/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.13.1.4.512 Lightweight name-value pair

Location: /formulation/[]/workflows/[]/properties/[]
Type: Object

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 436 — Properties for the property 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.

6.13.1.4.513 Name

Location: /formulation/[]/workflows/[]/properties/[]/name
Type: String

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

6.13.1.4.514 Value

Location: /formulation/[]/workflows/[]/properties/[]/value
Type: String

The value of the property.

© Ecma International 2025

6.13.1.5 Properties

Location: /formulation/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.13.1.5.1 Lightweight name-value pair

Location: /formulation/[]/properties/[]
Type: Object

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 437 — Properties for the property 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.

6.13.1.5.2 Name

Location: /formulation/[]/properties/[]/name
Type: String

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

6.13.1.5.3 Value

Location: /formulation/[]/properties/[]/value
Type: String

The value of the property.

6.14 Declarations

Location: /declarations
Property: declarations (Optional)
Type: Object

The list of declarations which describe the conformance to standards. Each declaration may include attestations, claims, and evidence.

© Ecma International 2025

Table 438 — 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.
signature	Array	Optional	Enveloped signature in JSON Signature Format (JSF).

6.14.1 Assessors

Location: /declarations/assessors
Property: assessors (Optional)
Type: Array

The list of assessors evaluating claims and determining conformance to requirements and confidence in that assessment. Each item of this array shall be an Assessor object.

6.14.1.1 Assessor

Location: /declarations/assessors/[]
Type: Object

The assessor who evaluates claims and determines conformance to requirements and confidence in that assessment.

Table 439 — Properties for the assessor object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM.
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.

6.14.1.1.1 BOM Reference

Location: /declarations/assessors/[]/bom-ref
Type: String

An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM.

© Ecma International 2025

6.14.1.1.2 Third Party

Location: /declarations/assessors/[]/thirdParty
Type: Boolean

The boolean indicating if the assessor is outside the organization generating claims. A value of false indicates a self assessor.

6.14.1.1.3 Organization

Location: /declarations/assessors/[]/organization
Type: Object

The entity issuing the assessment.

Table 440 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.14.1.1.4 BOM Reference

Location: /declarations/assessors/[]/organization/bom-ref
Type: String

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

6.14.1.1.5 Organization Name

Location: /declarations/assessors/[]/organization/name
Type: String

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.14.1.1.6 Organization Address

Location: /declarations/assessors/[]/organization/address
Type: Object

The physical address (location) of the organization.

© Ecma International 2025

Table 441 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.14.1.1.7 BOM Reference

Location: /declarations/assessors/[]/organization/address/bom-ref
Type: String

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

6.14.1.1.8 Country

Location: /declarations/assessors/[]/organization/address/country
Type: String

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

6.14.1.1.9 Region

Location: /declarations/assessors/[]/organization/address/region
Type: String

The region or state in the country.

Example (Informative)

```
Texas
```

6.14.1.1.10 Locality

Location: /declarations/assessors/[]/organization/address/locality
Type: String

The locality or city within the country.

Example (Informative)

```
Austin
```

© Ecma International 2025

6.14.1.1.11 Post Office Box Number

Location: /declarations/assessors/[]/organization/address/postOfficeBoxNumber
Type: String

The post office box number.

Example (Informative)
```
901
```

6.14.1.1.12 Postal Code

Location: /declarations/assessors/[]/organization/address/postalCode
Type: String

The postal code.

Example (Informative)
```
78758
```

6.14.1.1.13 Street Address

Location: /declarations/assessors/[]/organization/address/streetAddress
Type: String

The street address.

Example (Informative)
```
100 Main Street
```

6.14.1.1.14 Organization URL(s)

Location: /declarations/assessors/[]/organization/url
Property: url (Optional)
Type: Array (of String)
Format: iri-reference as specified in RFC 3987

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

Example (Informative)
```
https://example.com
```

6.14.1.1.15 Organizational Contact

Location: /declarations/assessors/[]/organization/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.14.1.1.16 Organizational Person

Location: /declarations/assessors/[]/organization/contact/[]
Type: Object

© Ecma International 2025

Table 442 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.14.1.1.17 BOM Reference

Location: /declarations/assessors/[]/organization/contact/[]/bom-ref
Type: String

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

6.14.1.1.18 Name

Location: /declarations/assessors/[]/organization/contact/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.14.1.1.19 Email Address

Location: /declarations/assessors/[]/organization/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

6.14.1.1.20 Phone

Location: /declarations/assessors/[]/organization/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)

```
800-555-1212
```

© Ecma International 2025

6.14.2 Attestations

Location: /declarations/attestations
Property: attestations (Optional)
Type: Array

The list of attestations asserted by an assessor that maps requirements to claims. Each item of this array shall be an Attestation object.

6.14.2.1 Attestation

Location: /declarations/attestations/[]
Type: Object

Table 443 — Properties for the attestation object

Property	Type	Requirement	Description
summary	String	Optional	The short description explaining the main points of the attestation.
assessor	String	Optional	The **`bom-ref`** to the assessor asserting the attestation.
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).

6.14.2.1.1 Summary

Location: /declarations/attestations/[]/summary
Type: String

The short description explaining the main points of the attestation.

6.14.2.1.2 Assessor

Location: /declarations/attestations/[]/assessor
Type: String

The **`bom-ref`** to the assessor asserting the attestation.

6.14.2.1.3 Map

Location: /declarations/attestations/[]/map
Property: map (Optional)
Type: Array

The grouping of requirements to claims and the attestors declared conformance and confidence thereof. Each item of this array shall be a Map object.

6.14.2.1.4 Map

Location: /declarations/attestations/[]/map/[]
Type: Object

© Ecma International 2025

Table 444 — Properties for the map object

Property	Type	Requirement	Description
requirement	String	Optional	The **bom-ref** to the requirement being attested to.
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.

6.14.2.1.5 Requirement

Location: /declarations/attestations/[]/map/[]/requirement
Type: String

The **bom-ref** to the requirement being attested to.

6.14.2.1.6 Claims

Location: /declarations/attestations/[]/map/[]/claims
Property: claims (Optional)
Type: Array (of String)

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 shall be a string.

6.14.2.1.7 Counter Claims

Location: /declarations/attestations/[]/map/[]/counterClaims
Property: counterClaims (Optional)
Type: Array (of String)

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 shall be a string.

6.14.2.1.8 Conformance

Location: /declarations/attestations/[]/map/[]/conformance
Type: Object

The conformance of the claim meeting a requirement.

Table 445 — 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.

© Ecma International 2025

6.14.2.1.9 Score

Location: /declarations/attestations/[]/map/[]/conformance/score
Type: Number
Maximum Value: 1

The conformance of the claim between and inclusive of 0 and 1, where 1 is 100% conformance.

6.14.2.1.10 Rationale

Location: /declarations/attestations/[]/map/[]/conformance/rationale
Type: String

The rationale for the conformance score.

6.14.2.1.11 Mitigation Strategies

Location: /declarations/attestations/[]/map/[]/conformance/mitigationStrategies
Property: mitigationStrategies (Optional)
Type: Array (of String)

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 shall be a string.

6.14.2.1.12 Confidence

Location: /declarations/attestations/[]/map/[]/confidence
Type: Object

The confidence of the claim meeting the requirement.

Table 446 — 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.

6.14.2.1.13 Score

Location: /declarations/attestations/[]/map/[]/confidence/score
Type: Number
Maximum Value: 1

The confidence of the claim between and inclusive of 0 and 1, where 1 is 100% confidence.

6.14.2.1.14 Rationale

Location: /declarations/attestations/[]/map/[]/confidence/rationale
Type: String

The rationale for the confidence score.

© Ecma International 2025

6.14.2.1.15 Signature

Location: /declarations/attestations/[]/signature
Type: Object

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.

6.14.3 Claims

Location: /declarations/claims
Property: claims (Optional)
Type: Array

The list of claims. Each item of this array shall be a Claim object.

6.14.3.1 Claim

Location: /declarations/claims/[]
Type: Object

Table 447 — Properties for the claim object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM.
target	String	Optional	The **`bom-ref`** to a target representing a specific system, application, API, module, team, person, process, business unit, company, etc... that this claim is being applied to.
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).

© Ecma International 2025

6.14.3.1.1 BOM Reference

Location: /declarations/claims/[]/bom-ref
Type: String

An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM.

6.14.3.1.2 Target

Location: /declarations/claims/[]/target
Type: String

The **`bom-ref`** to a target representing a specific system, application, API, module, team, person, process, business unit, company, etc... that this claim is being applied to.

6.14.3.1.3 Predicate

Location: /declarations/claims/[]/predicate
Type: String

The specific statement or assertion about the target.

6.14.3.1.4 Mitigation Strategies

Location: /declarations/claims/[]/mitigationStrategies
Property: mitigationStrategies (Optional)
Type: Array (of String)

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 shall be a string.

6.14.3.1.5 Reasoning

Location: /declarations/claims/[]/reasoning
Type: String

The written explanation of why the evidence provided substantiates the claim.

6.14.3.1.6 Evidence

Location: /declarations/claims/[]/evidence
Property: evidence (Optional)
Type: Array (of String)

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 shall be a string.

© Ecma International 2025

6.14.3.1.7 Counter Evidence

Location: /declarations/claims/[]/counterEvidence
Property: counterEvidence (Optional)
Type: Array (of String)

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 shall be a string.

6.14.3.1.8 External References

Location: /declarations/claims/[]/externalReferences
Property: externalReferences (Optional)
Type: Array

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 shall be an External Reference object.

6.14.3.1.9 External Reference

Location: /declarations/claims/[]/externalReferences/[]
Type: Object

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.

6.14.3.1.10 Signature

Location: /declarations/claims/[]/signature
Type: Object

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.

6.14.4 Evidence

Location: /declarations/evidence
Property: evidence (Optional)
Type: Array

The list of evidence Each item of this array shall be an Evidence object.

6.14.4.1 Evidence

Location: /declarations/evidence/[]
Type: Object

© Ecma International 2025

Table 448 — Properties for the evidence object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM.
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 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).

6.14.4.1.1 BOM Reference

Location: /declarations/evidence/[]/bom-ref
Type: String

An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM.

6.14.4.1.2 Property Name

Location: /declarations/evidence/[]/propertyName
Type: String

The reference to the property name as defined in the CycloneDX Property Taxonomy.

6.14.4.1.3 Description

Location: /declarations/evidence/[]/description
Type: String

The written description of what this evidence is and how it was created.

6.14.4.1.4 Data

Location: /declarations/evidence/[]/data
Property: data (Optional)
Type: Array

The output or analysis that supports claims. Each item of this array shall be a Data object.

© Ecma International 2025

6.14.4.1.5 Data

Location: /declarations/evidence/[]/data/[]
Type: Object

Table 449 — 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.

6.14.4.1.6 Data Name

Location: /declarations/evidence/[]/data/[]/name
Type: String

The name of the data.

6.14.4.1.7 Data Contents

Location: /declarations/evidence/[]/data/[]/contents
Type: Object

The contents or references to the contents of the data being described.

Table 450 — Properties for the contents object

Property	Type	Requirement	Description
attachment	Object	Optional	A way to include textual or encoded data.
url	String	Optional	The URL to where the data can be retrieved.

6.14.4.1.8 Data Attachment

Location: /declarations/evidence/[]/data/[]/contents/attachment
Type: Object

A way to include textual or encoded data.

© Ecma International 2025

Table 451 — 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 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.

6.14.4.1.9 Content-Type

Location: /declarations/evidence/[]/data/[]/contents/attachment/contentType
Type: String
Default Value: text/plain

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.

Example 1 (Informative)

```
text/plain
```

Example 2 (Informative)

```
application/json
```

Example 3 (Informative)

```
image/png
```

6.14.4.1.10 Encoding

Location: /declarations/evidence/[]/data/[]/contents/attachment/encoding
Type: String (enum)

Specifies the encoding the text is represented in.

Table 452 — Enumeration of possible values

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

6.14.4.1.11 Attachment Text

Location: /declarations/evidence/[]/data/[]/contents/attachment/content
Type: String

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

© Ecma International 2025

6.14.4.1.12 Data URL

Location: /declarations/evidence/[]/data/[]/contents/url
Type: String
Format: iri-reference as specified in RFC 3987

The URL to where the data can be retrieved.

6.14.4.1.13 Data Classification

Location: /declarations/evidence/[]/data/[]/classification
Type: String

Data classification tags data according to its type, sensitivity, and value if altered, stolen, or destroyed.

6.14.4.1.14 Sensitive Data

Location: /declarations/evidence/[]/data/[]/sensitiveData
Property: sensitiveData (Optional)
Type: Array (of String)

A description of any sensitive data included. Each item of this array shall be a string.

6.14.4.1.15 Data Governance

Location: /declarations/evidence/[]/data/[]/governance
Type: Object

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 453 — 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.

6.14.4.1.16 Data Custodians

Location: /declarations/evidence/[]/data/[]/governance/custodians
Property: custodians (Optional)
Type: Array

Data custodians are responsible for the safe custody, transport, and storage of data.

6.14.4.1.17 Custodian

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]
Type: Object

© Ecma International 2025

Table 454 — Properties for the custodian 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).

6.14.4.1.18 Organization

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization
Type: Object

The organization that is responsible for specific data governance role(s).

Table 455 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.14.4.1.19 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/bom-ref
Type: String

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

6.14.4.1.20 Organization Name

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/name
Type: String

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.14.4.1.21 Organization Address

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/address
Type: Object

The physical address (location) of the organization.

© Ecma International 2025

Table 456 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.14.4.1.22 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/address/bom-ref
Type: String

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

6.14.4.1.23 Country

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/address/country
Type: String

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

6.14.4.1.24 Region

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/address/region
Type: String

The region or state in the country.

Example (Informative)

```
Texas
```

6.14.4.1.25 Locality

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/address/locality
Type: String

The locality or city within the country.

Example (Informative)

```
Austin
```

© Ecma International 2025

6.14.4.1.26 Post Office Box Number

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/address/postOfficeBoxNumber
Type: String

The post office box number.

Example (Informative)
```
901
```

6.14.4.1.27 Postal Code

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/address/postalCode
Type: String

The postal code.

Example (Informative)
```
78758
```

6.14.4.1.28 Street Address

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/address/streetAddress
Type: String

The street address.

Example (Informative)
```
100 Main Street
```

6.14.4.1.29 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

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

Example (Informative)
```
https://example.com
```

6.14.4.1.30 Organizational Contact

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.14.4.1.31 Organizational Person

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/contact/[]
Type: Object

© Ecma International 2025

Table 457 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.14.4.1.32 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/contact/[]/bom-ref
Type: String

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

6.14.4.1.33 Name

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/contact/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.14.4.1.34 Email Address

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

6.14.4.1.35 Phone

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/organization/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)

```
800-555-1212
```

© Ecma International 2025

6.14.4.1.36 Individual

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/contact
Type: Object

The individual that is responsible for specific data governance role(s).

Table 458 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.14.4.1.37 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/contact/bom-ref
Type: String

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

6.14.4.1.38 Name

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/contact/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.14.4.1.39 Email Address

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/contact/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```


© Ecma International 2025

6.14.4.1.40 Phone

Location: /declarations/evidence/[]/data/[]/governance/custodians/[]/contact/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.14.4.1.41 Data Stewards

Location: /declarations/evidence/[]/data/[]/governance/stewards
Property: stewards (Optional)
Type: Array

Data stewards are responsible for data content, context, and associated business rules.

6.14.4.1.42 Steward

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]
Type: Object

Table 459 — Properties for the steward 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).

6.14.4.1.43 Organization

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization
Type: Object

The organization that is responsible for specific data governance role(s).

Table 460 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

© Ecma International 2025

6.14.4.1.44 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/bom-ref
Type: String

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

6.14.4.1.45 Organization Name

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/name
Type: String

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.14.4.1.46 Organization Address

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/address
Type: Object

The physical address (location) of the organization.

Table 461 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.14.4.1.47 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/address/bom-ref
Type: String

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

© Ecma International 2025

6.14.4.1.48 Country

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/address/country
Type: String

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

6.14.4.1.49 Region

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/address/region
Type: String

The region or state in the country.

Example (Informative)

```
Texas
```

6.14.4.1.50 Locality

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/address/locality
Type: String

The locality or city within the country.

Example (Informative)

```
Austin
```

6.14.4.1.51 Post Office Box Number

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/address/postOfficeBoxNumber
Type: String

The post office box number.

Example (Informative)

```
901
```

6.14.4.1.52 Postal Code

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/address/postalCode
Type: String

The postal code.

Example (Informative)

```
78758
```

© Ecma International 2025

6.14.4.1.53 Street Address

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/address/streetAddress
Type: String

The street address.

Example (Informative)

```
100 Main Street
```

6.14.4.1.54 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

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

Example (Informative)

```
https://example.com
```

6.14.4.1.55 Organizational Contact

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.14.4.1.56 Organizational Person

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/contact/[]
Type: Object

Table 462 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.14.4.1.57 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/contact/[]/bom-ref
Type: String

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

© Ecma International 2025

6.14.4.1.58 Name

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/contact/[]/name
Type: String

The name of a contact.

Example (Informative)
```
Contact name
```

6.14.4.1.59 Email Address

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.14.4.1.60 Phone

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/organization/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.14.4.1.61 Individual

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/contact
Type: Object

The individual that is responsible for specific data governance role(s).

Table 463 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

© Ecma International 2025

6.14.4.1.62 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/contact/bom-ref
Type: String

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

6.14.4.1.63 Name

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/contact/name
Type: String

The name of a contact.

Example (Informative)
```
Contact name
```

6.14.4.1.64 Email Address

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/contact/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.14.4.1.65 Phone

Location: /declarations/evidence/[]/data/[]/governance/stewards/[]/contact/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.14.4.1.66 Data Owners

Location: /declarations/evidence/[]/data/[]/governance/owners
Property: owners (Optional)
Type: Array

Data owners are concerned with risk and appropriate access to data.

6.14.4.1.67 Owner

Location: /declarations/evidence/[]/data/[]/governance/owners/[]
Type: Object

© Ecma International 2025

Table 464 — Properties for the owner 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).

6.14.4.1.68 Organization

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization
Type: Object

The organization that is responsible for specific data governance role(s).

Table 465 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.14.4.1.69 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/bom-ref
Type: String

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

6.14.4.1.70 Organization Name

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/name
Type: String

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.14.4.1.71 Organization Address

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/address
Type: Object

The physical address (location) of the organization.

© Ecma International 2025

Table 466 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.14.4.1.72 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/address/bom-ref
Type: String

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

6.14.4.1.73 Country

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/address/country
Type: String

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

6.14.4.1.74 Region

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/address/region
Type: String

The region or state in the country.

Example (Informative)

```
Texas
```

6.14.4.1.75 Locality

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/address/locality
Type: String

The locality or city within the country.

Example (Informative)

```
Austin
```


© Ecma International 2025

6.14.4.1.76 Post Office Box Number

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/address/postOfficeBoxNumber
Type: String

The post office box number.

Example (Informative)
```
901
```

6.14.4.1.77 Postal Code

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/address/postalCode
Type: String

The postal code.

Example (Informative)
```
78758
```

6.14.4.1.78 Street Address

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/address/streetAddress
Type: String

The street address.

Example (Informative)
```
100 Main Street
```

6.14.4.1.79 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

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

Example (Informative)
```
https://example.com
```

6.14.4.1.80 Organizational Contact

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.14.4.1.81 Organizational Person

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/contact/[]
Type: Object

© Ecma International 2025

Table 467 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.14.4.1.82 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/contact/[]/bom-ref
Type: String

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

6.14.4.1.83 Name

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/contact/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.14.4.1.84 Email Address

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

6.14.4.1.85 Phone

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/organization/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)

```
800-555-1212
```


© Ecma International 2025

6.14.4.1.86 Individual

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/contact
Type: Object

The individual that is responsible for specific data governance role(s).

Table 468 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.14.4.1.87 BOM Reference

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/contact/bom-ref
Type: String

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

6.14.4.1.88 Name

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/contact/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.14.4.1.89 Email Address

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/contact/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

© Ecma International 2025

6.14.4.1.90 Phone

Location: /declarations/evidence/[]/data/[]/governance/owners/[]/contact/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.14.4.1.91 Created

Location: /declarations/evidence/[]/created
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the evidence was created.

6.14.4.1.92 Expires

Location: /declarations/evidence/[]/expires
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time (timestamp) when the evidence is no longer valid.

6.14.4.1.93 Author

Location: /declarations/evidence/[]/author
Type: Object

The author of the evidence.

Table 469 — Properties for the author object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.14.4.1.94 BOM Reference

Location: /declarations/evidence/[]/author/bom-ref
Type: String

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

© Ecma International 2025

6.14.4.1.95 Name

Location: /declarations/evidence/[]/author/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.14.4.1.96 Email Address

Location: /declarations/evidence/[]/author/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

6.14.4.1.97 Phone

Location: /declarations/evidence/[]/author/phone
Type: String

The phone number of the contact.

Example (Informative)

```
800-555-1212
```

6.14.4.1.98 Reviewer

Location: /declarations/evidence/[]/reviewer
Type: Object

The reviewer of the evidence.

Table 470 — Properties for the reviewer object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

© Ecma International 2025

6.14.4.1.99 BOM Reference

Location: /declarations/evidence/[]/reviewer/bom-ref
Type: String

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

6.14.4.1.100 Name

Location: /declarations/evidence/[]/reviewer/name
Type: String

The name of a contact.

Example (Informative)
```
Contact name
```

6.14.4.1.101 Email Address

Location: /declarations/evidence/[]/reviewer/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.14.4.1.102 Phone

Location: /declarations/evidence/[]/reviewer/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.14.4.1.103 Signature

Location: /declarations/evidence/[]/signature
Type: Object

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.

© Ecma International 2025

6.14.5 Targets

Location: /declarations/targets
Property: targets (Optional)
Type: Object

The list of targets which claims are made against.

Table 471 — 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.

6.14.5.1 Organizations

Location: /declarations/targets/organizations
Property: organizations (Optional)
Type: Array

The list of organizations which claims are made against.

6.14.5.1.1 Organization

Location: /declarations/targets/organizations/[]
Type: Object

Table 472 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.14.5.1.2 BOM Reference

Location: /declarations/targets/organizations/[]/bom-ref
Type: String

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

© Ecma International 2025

6.14.5.1.3 Organization Name

Location: /declarations/targets/organizations/[]/name
Type: String

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.14.5.1.4 Organization Address

Location: /declarations/targets/organizations/[]/address
Type: Object

The physical address (location) of the organization.

Table 473 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.14.5.1.5 BOM Reference

Location: /declarations/targets/organizations/[]/address/bom-ref
Type: String

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

6.14.5.1.6 Country

Location: /declarations/targets/organizations/[]/address/country
Type: String

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

© Ecma International 2025

6.14.5.1.7 Region

Location: /declarations/targets/organizations/[]/address/region
Type: String

The region or state in the country.

Example (Informative)
```
Texas
```

6.14.5.1.8 Locality

Location: /declarations/targets/organizations/[]/address/locality
Type: String

The locality or city within the country.

Example (Informative)
```
Austin
```

6.14.5.1.9 Post Office Box Number

Location: /declarations/targets/organizations/[]/address/postOfficeBoxNumber
Type: String

The post office box number.

Example (Informative)
```
901
```

6.14.5.1.10 Postal Code

Location: /declarations/targets/organizations/[]/address/postalCode
Type: String

The postal code.

Example (Informative)
```
78758
```

6.14.5.1.11 Street Address

Location: /declarations/targets/organizations/[]/address/streetAddress
Type: String

The street address.

Example (Informative)
```
100 Main Street
```

© Ecma International 2025

6.14.5.1.12 Organization URL(s)

Location: /declarations/targets/organizations/[]/url
Property: url (Optional)
Type: Array (of String)
Format: iri-reference as specified in RFC 3987

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

Example (Informative)

```
https://example.com
```

6.14.5.1.13 Organizational Contact

Location: /declarations/targets/organizations/[]/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.14.5.1.14 Organizational Person

Location: /declarations/targets/organizations/[]/contact/[]
Type: Object

Table 474 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.14.5.1.15 BOM Reference

Location: /declarations/targets/organizations/[]/contact/[]/bom-ref
Type: String

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

6.14.5.1.16 Name

Location: /declarations/targets/organizations/[]/contact/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```


© Ecma International 2025

6.14.5.1.17 Email Address

Location: /declarations/targets/organizations/[]/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.14.5.1.18 Phone

Location: /declarations/targets/organizations/[]/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.14.5.2 Components

Location: /declarations/targets/components
Property: components (Optional)
Type: Array

The list of components which claims are made against. Each item of this array shall be a Component object.

6.14.5.2.1 Component

Location: /declarations/targets/components/[]
Type: Object

6.14.5.3 Services

Location: /declarations/targets/services
Property: services (Optional)
Type: Array

The list of services which claims are made against. Each item of this array shall be a Service object.

6.14.5.3.1 Service

Location: /declarations/targets/services/[]
Type: Object

© Ecma International 2025

6.14.6 Affirmation

Location: /declarations/affirmation
Property: affirmation (Optional)
Type: Object

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 475 — 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).

6.14.6.1 Statement

Location: /declarations/affirmation/statement
Property: statement (Optional)
Type: String

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.

Example (Informative)

```
I certify, to the best of my knowledge, that all information is correct.
```

6.14.6.2 Signatories

Location: /declarations/affirmation/signatories
Property: signatories (Optional)
Type: Array

The list of signatories authorized on behalf of an organization to assert validity of this document. Each item of this array shall be a Signatory object.

6.14.6.2.1 Signatory

Location: /declarations/affirmation/signatories/[]
Type: Object

© Ecma International 2025

Table 476 — Properties for the signatory 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.

6.14.6.2.2 Name

Location: /declarations/affirmation/signatories/[]/name
Type: String

The signatory's name.

6.14.6.2.3 Role

Location: /declarations/affirmation/signatories/[]/role
Type: String

The signatory's role within an organization.

6.14.6.2.4 Signature

Location: /declarations/affirmation/signatories/[]/signature
Type: Object

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.

6.14.6.2.5 Organization

Location: /declarations/affirmation/signatories/[]/organization
Type: Object

The signatory's organization.

© Ecma International 2025

Table 477 — Properties for the organization object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.14.6.2.6 BOM Reference

Location: /declarations/affirmation/signatories/[]/organization/bom-ref
Type: String

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

6.14.6.2.7 Organization Name

Location: /declarations/affirmation/signatories/[]/organization/name
Type: String

The name of the organization.

Example (Informative)

```
Example Inc.
```

6.14.6.2.8 Organization Address

Location: /declarations/affirmation/signatories/[]/organization/address
Type: Object

The physical address (location) of the organization.

Table 478 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

© Ecma International 2025

Table 478 — Properties for the address object *(continued)*

Property	Type	Requirement	Description
postalCode	String	Optional	The postal code.
streetAddress	String	Optional	The street address.

6.14.6.2.9 BOM Reference

Location: /declarations/affirmation/signatories/[]/organization/address/bom-ref
Type: String

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

6.14.6.2.10 Country

Location: /declarations/affirmation/signatories/[]/organization/address/country
Type: String

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

6.14.6.2.11 Region

Location: /declarations/affirmation/signatories/[]/organization/address/region
Type: String

The region or state in the country.

Example (Informative)

```
Texas
```

6.14.6.2.12 Locality

Location: /declarations/affirmation/signatories/[]/organization/address/locality
Type: String

The locality or city within the country.

Example (Informative)

```
Austin
```

6.14.6.2.13 Post Office Box Number

Location: /declarations/affirmation/signatories/[]/organization/address/postOfficeBoxNumber
Type: String

The post office box number.

Example (Informative)

```
901
```

© Ecma International 2025

6.14.6.2.14 Postal Code

Location: /declarations/affirmation/signatories/[]/organization/address/postalCode
Type: String

The postal code.

Example (Informative)
```
78758
```

6.14.6.2.15 Street Address

Location: /declarations/affirmation/signatories/[]/organization/address/streetAddress
Type: String

The street address.

Example (Informative)
```
100 Main Street
```

6.14.6.2.16 Organization URL(s)

Location: /declarations/affirmation/signatories/[]/organization/url
Property: url (Optional)
Type: Array (of String)
Format: iri-reference as specified in RFC 3987

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

Example (Informative)
```
https://example.com
```

6.14.6.2.17 Organizational Contact

Location: /declarations/affirmation/signatories/[]/organization/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.14.6.2.18 Organizational Person

Location: /declarations/affirmation/signatories/[]/organization/contact/[]
Type: Object

© Ecma International 2025

Table 479 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.14.6.2.19 BOM Reference

Location: /declarations/affirmation/signatories/[]/organization/contact/[]/bom-ref
Type: String

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

6.14.6.2.20 Name

Location: /declarations/affirmation/signatories/[]/organization/contact/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.14.6.2.21 Email Address

Location: /declarations/affirmation/signatories/[]/organization/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

6.14.6.2.22 Phone

Location: /declarations/affirmation/signatories/[]/organization/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)

```
800-555-1212
```

© Ecma International 2025

6.14.6.2.23 External Reference

Location: /declarations/affirmation/signatories/[]/externalReference
Type: Object

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.

6.14.6.3 Signature

Location: /declarations/affirmation/signature
Property: signature (Optional)
Type: Object

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.

6.14.7 Signature

Location: /declarations/signature
Property: signature (Optional)
Type: Object

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.

6.15 Definitions

Location: /definitions
Property: definitions (Optional)
Type: Object

A collection of reusable objects that are defined and may be used elsewhere in the BOM.

Table 480 — 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.
patents	Array	Optional	The list of either individual patents or patent families.

© Ecma International 2025

6.15.1 Standards

Location: /definitions/standards
Property: standards (Optional)
Type: Array

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. Each item of this array shall be a Standard object.

6.15.1.1 Standard

Location: /definitions/standards/[]
Type: Object

A standard 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.

Table 481 — Properties for the standard object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM.
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.
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).

6.15.1.1.1 BOM Reference

Location: /definitions/standards/[]/bom-ref
Type: String

An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM.

© Ecma International 2025

6.15.1.1.2 Name

Location: /definitions/standards/[]/name
Type: String

The name of the standard. This will often be a shortened, single name of the standard.

6.15.1.1.3 Version

Location: /definitions/standards/[]/version
Type: String

The version of the standard.

6.15.1.1.4 Description

Location: /definitions/standards/[]/description
Type: String

The description of the standard.

6.15.1.1.5 Owner

Location: /definitions/standards/[]/owner
Type: String

The owner of the standard, often the entity responsible for its release.

6.15.1.1.6 Requirements

Location: /definitions/standards/[]/requirements
Property: requirements (Optional)
Type: Array

The list of requirements comprising the standard. Each item of this array shall be a Requirement object.

6.15.1.1.7 Requirement

Location: /definitions/standards/[]/requirements/[]
Type: Object

Table 482 — Properties for the requirement object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM.
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.

© Ecma International 2025

Table 482 — Properties for the requirement object *(continued)*

Property	Type	Requirement	Description
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	The **`bom-ref`** to a parent requirement. This establishes a hierarchy of requirements. Top-level requirements shall not define a parent. Only child requirements should define parents.
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.

6.15.1.1.8 BOM Reference

Location: /definitions/standards/[]/requirements/[]/bom-ref
Type: String

An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM.

6.15.1.1.9 Identifier

Location: /definitions/standards/[]/requirements/[]/identifier
Type: String

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.

6.15.1.1.10 Title

Location: /definitions/standards/[]/requirements/[]/title
Type: String

The title of the requirement.

© Ecma International 2025

6.15.1.1.11 Text

Location: /definitions/standards/[]/requirements/[]/text
Type: String

The textual content of the requirement.

6.15.1.1.12 Descriptions

Location: /definitions/standards/[]/requirements/[]/descriptions
Property: descriptions (Optional)
Type: Array (of String)

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 shall be a string.

6.15.1.1.13 OWASP OpenCRE Identifier(s)

Location: /definitions/standards/[]/requirements/[]/openCre
Property: openCre (Optional)
Type: Array (of String)
Pattern Constraint: `^CRE:[0-9]+-[0-9]+$`

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 shall be a string.

Example (Informative)
`CRE:764-507`

6.15.1.1.14 Parent BOM Reference

Location: /definitions/standards/[]/requirements/[]/parent
Type: String

The **`bom-ref`** to a parent requirement. This establishes a hierarchy of requirements. Top-level requirements shall not define a parent. Only child requirements should define parents.

6.15.1.1.15 Properties

Location: /definitions/standards/[]/requirements/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

© Ecma International 2025

6.15.1.1.16 Lightweight name-value pair

Location: /definitions/standards/[]/requirements/[]/properties/[]
Type: Object

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 483 — Properties for the property 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.

6.15.1.1.17 Name

Location: /definitions/standards/[]/requirements/[]/properties/[]/name
Type: String

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

6.15.1.1.18 Value

Location: /definitions/standards/[]/requirements/[]/properties/[]/value
Type: String

The value of the property.

6.15.1.1.19 External References

Location: /definitions/standards/[]/requirements/[]/externalReferences
Property: externalReferences (Optional)
Type: Array

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 shall be an External Reference object.

6.15.1.1.20 External Reference

Location: /definitions/standards/[]/requirements/[]/externalReferences/[]
Type: Object

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.

© Ecma International 2025

6.15.1.1.21 Levels

Location: /definitions/standards/[]/levels
Property: levels (Optional)
Type: Array

The list of levels associated with the standard. Some standards have different levels of compliance. Each item of this array shall be a Level object.

6.15.1.1.22 Level

Location: /definitions/standards/[]/levels/[]
Type: Object

Table 484 — Properties for the level object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM.
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-ref`**s that comprise the level.

6.15.1.1.23 BOM Reference

Location: /definitions/standards/[]/levels/[]/bom-ref
Type: String

An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM.

6.15.1.1.24 Identifier

Location: /definitions/standards/[]/levels/[]/identifier
Type: String

The identifier used in the standard to identify a specific level.

6.15.1.1.25 Title

Location: /definitions/standards/[]/levels/[]/title
Type: String

The title of the level.

6.15.1.1.26 Description

Location: /definitions/standards/[]/levels/[]/description
Type: String

The description of the level.

© Ecma International 2025

6.15.1.1.27 Requirements

Location: /definitions/standards/[]/levels/[]/requirements
Property: requirements (Optional)
Type: Array (of String)

The list of requirement **`bom-ref`**s 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 shall be a string.

6.15.1.1.28 External References

Location: /definitions/standards/[]/externalReferences
Property: externalReferences (Optional)
Type: Array

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 shall be an External Reference object.

6.15.1.1.29 External Reference

Location: /definitions/standards/[]/externalReferences/[]
Type: Object

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.

6.15.1.1.30 Signature

Location: /definitions/standards/[]/signature
Type: Object

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.

6.15.2 Patents

Location: /definitions/patents
Property: patents (Optional)
Type: Array

The list of either individual patents or patent families.

6.15.2.1 Patent

Location: /definitions/patents/[]

Shall be any of:

1. Patent
2. Patent Family

© Ecma International 2025

6.15.2.1.1 Patent

Type: Object

A patent is a legal instrument, granted by an authority, that confers certain rights over an invention for a specified period, contingent on public disclosure and adherence to relevant legal requirements. The summary information in this object is aligned with WIPO ST.96 principles where applicable.

Table 485 — Properties for the patent object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM.
patentNumber	String	Required	The unique number assigned to the granted patent by the issuing authority. Aligned with **`PatentNumber`** in WIPO ST.96. Refer to PatentNumber in ST.96.
applicationNumber	String	Optional	The unique number assigned to a patent application when it is filed with a patent office. It is used to identify the specific application and track its progress through the examination process. Aligned with **`ApplicationNumber`** in ST.96. Refer to ApplicationIdentificationType in ST.96.
jurisdiction	String	Required	The jurisdiction or patent office where the priority application was filed, specified using WIPO ST.3 codes. Aligned with **`IPOfficeCode`** in ST.96. Refer to IPOfficeCode in ST.96.
priorityApplication	Object	Optional	The priorityApplication contains the essential data necessary to identify and reference an earlier patent filing for priority rights. In line with WIPO ST.96 guidelines, it includes the jurisdiction (office code), application number, and filing date- the three key elements that uniquely specify the priority application in a global patent context.
publicationNumber	String	Optional	This is the number assigned to a patent application once it is published. Patent applications are generally published 18 months after filing (unless an applicant requests non-publication). This number is distinct from the application number. Purpose: Identifies the publicly available version of the application. Format: Varies by jurisdiction, often similar to application numbers but includes an additional suffix indicating publication. Example: - US: US20240000123A1 (indicates the first publication of application US20240000123) - Europe: EP23123456A1 (first publication of European application EP23123456). WIPO ST.96 v8.0: - Publication Number field: https://www.wipo.int/standards/XMLSchema/ST96/V8_0/Patent/PublicationNumber.xsd.
title	String	Optional	The title of the patent, summarising the invention it protects. Aligned with **`InventionTitle`** in WIPO ST.96. Refer to InventionTitle in ST.96.
abstract	String	Optional	A brief summary of the invention described in the patent. Aligned with **`Abstract`** and **`P`** in WIPO ST.96. Refer to Abstract in ST.96.
filingDate	String	Optional	The date the patent application was filed with the jurisdiction. Aligned with **`FilingDate`** in WIPO ST.96. Refer to FilingDate in ST.96.

© Ecma International 2025

Table 485 — Properties for the patent object *(continued)*

Property	Type	Requirement	Description
grantDate	String	Optional	The date the patent was granted by the jurisdiction. Aligned with **`GrantDate`** in WIPO ST.96. Refer to GrantDate in ST.96.
patentExpirationDate	String	Optional	The date the patent expires. Derived from grant or filing date according to jurisdiction-specific rules.
patentLegalStatus	String	Required	Indicates the current legal status of the patent or patent application, based on the WIPO ST.27 standard. This status reflects administrative, procedural, or legal events. Values include both active and inactive states and are useful for determining enforceability, procedural history, and maintenance status.
patentAssignee	Array	Optional	A collection of organisations or individuals to whom the patent rights are assigned. This supports joint ownership and allows for flexible representation of both corporate entities and individual inventors.
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.

6.15.2.1.2 Patent Family

Type: Object

A patent family is a group of related patent applications or granted patents that cover the same or similar invention. These patents are filed in multiple jurisdictions to protect the invention across different regions or countries. A patent family typically includes patents that share a common priority date, originating from the same initial application, and may vary slightly in scope or claims to comply with regional legal frameworks. Fields align with WIPO ST.96 standards where applicable.

Table 486 — Properties for the patent family object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. For a patent, it might be a good idea to use a patent number as the BOM reference ID.
familyId	String	Required	The unique identifier for the patent family, aligned with the **`id`** attribute in WIPO ST.96 v8.0's **`PatentFamilyType`**. Refer to PatentFamilyType in ST.96.
priorityApplication	Object	Optional	The priorityApplication contains the essential data necessary to identify and reference an earlier patent filing for priority rights. In line with WIPO ST.96 guidelines, it includes the jurisdiction (office code), application number, and filing date-the three key elements that uniquely specify the priority application in a global patent context.

© Ecma International 2025

Table 486 — Properties for the patent family object *(continued)*

Property	Type	Requirement	Description
members	Array	Optional	A collection of patents or applications that belong to this family, each identified by a **`bom-ref`** pointing to a patent object defined elsewhere in the BOM.
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.

6.15.2.1.3 BOM Reference

Location: /definitions/patents/[]/bom-ref
Property: bom-ref (Optional)
Type: String

An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM.

6.15.2.1.4 Patent Number

Location: /definitions/patents/[]/patentNumber
Property: patentNumber (Required)
Type: String
Pattern Constraint: `^[A-Za-z0-9][A-Za-z0-9\-/.()\s]{0,28}[A-Za-z0-9]$`

The unique number assigned to the granted patent by the issuing authority. Aligned with **`PatentNumber`** in WIPO ST.96. Refer to PatentNumber in ST.96.

Example 1 (Informative)
`US987654321`

Example 2 (Informative)
`EP1234567B1`

6.15.2.1.5 Patent Application Number

Location: /definitions/patents/[]/applicationNumber
Property: applicationNumber (Optional)
Type: String
Pattern Constraint: `^[A-Za-z0-9][A-Za-z0-9\-/.()\s]{0,28}[A-Za-z0-9]$`

The unique number assigned to a patent application when it is filed with a patent office. It is used to identify the specific application and track its progress through the examination process. Aligned with **`ApplicationNumber`** in ST.96. Refer to ApplicationIdentificationType in ST.96.

Example 1 (Informative)
`US20240000123`

Example 2 (Informative)
`EP23123456`

© Ecma International 2025

6.15.2.1.6 Jurisdiction

Location: /definitions/patents/[]/jurisdiction
Property: jurisdiction (Required)
Type: String
Pattern Constraint: `^[A-Z]{2}$`

The jurisdiction or patent office where the priority application was filed, specified using WIPO ST.3 codes. Aligned with **`IPOfficeCode`** in ST.96. Refer to IPOfficeCode in ST.96.

Example 1 (Informative)

```
US
```

Example 2 (Informative)

```
EP
```

Example 3 (Informative)

```
JP
```

6.15.2.1.7 Priority Application

Location: /definitions/patents/[]/priorityApplication
Property: priorityApplication (Optional)
Type: Object

The priorityApplication contains the essential data necessary to identify and reference an earlier patent filing for priority rights. In line with WIPO ST.96 guidelines, it includes the jurisdiction (office code), application number, and filing date-the three key elements that uniquely specify the priority application in a global patent context.

Table 487 — Properties for the priorityApplication object

Property	Type	Requirement	Description
applicationNumber	String	Required	The unique number assigned to a patent application when it is filed with a patent office. It is used to identify the specific application and track its progress through the examination process. Aligned with **`ApplicationNumber`** in ST.96. Refer to ApplicationIdentificationType in ST.96.
jurisdiction	String	Required	The jurisdiction or patent office where the priority application was filed, specified using WIPO ST.3 codes. Aligned with **`IPOfficeCode`** in ST.96. Refer to IPOfficeCode in ST.96.
filingDate	String	Required	The date the priority application was filed, aligned with **`FilingDate`** in ST.96. Refer to FilingDate in ST.96.

6.15.2.1.8 Patent Application Number

Location: /definitions/patents/[]/priorityApplication/applicationNumber
Property: applicationNumber (Required)
Type: String
Pattern Constraint: `^[A-Za-z0-9][A-Za-z0-9\-/.()\s]{0,28}[A-Za-z0-9]$`

The unique number assigned to a patent application when it is filed with a patent office. It is used to identify the specific application and track its progress through the examination process. Aligned with **`ApplicationNumber`** in ST.96. Refer to ApplicationIdentificationType in ST.96.

© Ecma International 2025

Example 1 (Informative)
US20240000123

Example 2 (Informative)
EP23123456

6.15.2.1.9 Jurisdiction

Location: /definitions/patents/[]/priorityApplication/jurisdiction
Property: jurisdiction (Required)
Type: String
Pattern Constraint: ^[A-Z]{2}$

The jurisdiction or patent office where the priority application was filed, specified using WIPO ST.3 codes. Aligned with **IPOfficeCode** in ST.96. Refer to IPOfficeCode in ST.96.

Example 1 (Informative)
US

Example 2 (Informative)
EP

Example 3 (Informative)
JP

6.15.2.1.10 Filing Date

Location: /definitions/patents/[]/priorityApplication/filingDate
Property: filingDate (Required)
Type: String
Format: date as specified in RFC 3339 section 5.6

The date the priority application was filed, aligned with **FilingDate** in ST.96. Refer to FilingDate in ST.96.

6.15.2.1.11 Patent Publication Number

Location: /definitions/patents/[]/publicationNumber
Property: publicationNumber (Optional)
Type: String
Pattern Constraint: ^[A-Za-z0-9][A-Za-z0-9\-/.()\s]{0,28}[A-Za-z0-9]$

This is the number assigned to a patent application once it is published. Patent applications are generally published 18 months after filing (unless an applicant requests non-publication). This number is distinct from the application number. Purpose: Identifies the publicly available version of the application. Format: Varies by jurisdiction, often similar to application numbers but includes an additional suffix indicating publication. Example: - US: US20240000123A1 (indicates the first publication of application US20240000123) - Europe: EP23123456A1 (first publication of European application EP23123456). WIPO ST.96 v8.0: - Publication Number field: https://www.wipo.int/standards/XMLSchema/ST96/V8_0/Patent/PublicationNumber.xsd.

© Ecma International 2025

6.15.2.1.12 Patent Title

Location: /definitions/patents/[]/title
Property: title (Optional)
Type: String

The title of the patent, summarising the invention it protects. Aligned with **`InventionTitle`** in WIPO ST.96. Refer to InventionTitle in ST.96.

6.15.2.1.13 Patent Abstract

Location: /definitions/patents/[]/abstract
Property: abstract (Optional)
Type: String

A brief summary of the invention described in the patent. Aligned with **`Abstract`** and **`P`** in WIPO ST.96. Refer to Abstract in ST.96.

6.15.2.1.14 Filing Date

Location: /definitions/patents/[]/filingDate
Property: filingDate (Optional)
Type: String
Format: date as specified in RFC 3339 section 5.6

The date the patent application was filed with the jurisdiction. Aligned with **`FilingDate`** in WIPO ST.96. Refer to FilingDate in ST.96.

6.15.2.1.15 Grant Date

Location: /definitions/patents/[]/grantDate
Property: grantDate (Optional)
Type: String
Format: date as specified in RFC 3339 section 5.6

The date the patent was granted by the jurisdiction. Aligned with **`GrantDate`** in WIPO ST.96. Refer to GrantDate in ST.96.

6.15.2.1.16 Expiration Date

Location: /definitions/patents/[]/patentExpirationDate
Property: patentExpirationDate (Optional)
Type: String
Format: date as specified in RFC 3339 section 5.6

The date the patent expires. Derived from grant or filing date according to jurisdiction-specific rules.

6.15.2.1.17 Legal Status

Location: /definitions/patents/[]/patentLegalStatus
Property: patentLegalStatus (Required)
Type: String (enum)

Indicates the current legal status of the patent or patent application, based on the WIPO ST.27 standard. This status reflects administrative, procedural, or legal events. Values include both active and inactive states and are useful for determining enforceability, procedural history, and maintenance status.

© Ecma International 2025

Table 488 — Enumeration of possible values

Value	Description
pending	The patent application has been filed but not yet examined or granted.
granted	The patent application has been examined and a patent has been issued.
revoked	The patent has been declared invalid through a legal or administrative process.
expired	The patent has reached the end of its enforceable term.
lapsed	The patent is no longer in force due to non-payment of maintenance fees or other requirements.
withdrawn	The patent application was voluntarily withdrawn by the applicant.
abandoned	The patent application was abandoned, often due to lack of action or response.
suspended	Processing of the patent application has been temporarily halted.
reinstated	A previously abandoned or lapsed patent has been reinstated.
opposed	The patent application or granted patent is under formal opposition proceedings.
terminated	The patent or application has been officially terminated.
invalidated	The patent has been invalidated, either in part or in full.
in-force	The granted patent is active and enforceable.

6.15.2.1.18 Patent Assignees

Location: /definitions/patents/[]/patentAssignee
Property: patentAssignee (Optional)
Type: Array

A collection of organisations or individuals to whom the patent rights are assigned. This supports joint ownership and allows for flexible representation of both corporate entities and individual inventors.

6.15.2.1.19 PatentAssignee

Location: /definitions/patents/[]/patentAssignee/[]

Shall be one of:

1. Person
2. Organizational Entity

6.15.2.1.20 Person

Type: Object

© Ecma International 2025

Table 489 — Properties for the person object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.15.2.1.21 Organizational Entity

Type: Object

Table 490 — Properties for the organizational entity object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.15.2.1.22 BOM Reference

Location: /definitions/patents/[]/patentAssignee/[]/bom-ref
Property: bom-ref (Optional)
Type: String

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

6.15.2.1.23 Name

Location: /definitions/patents/[]/patentAssignee/[]/name
Property: name (Optional)
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

© Ecma International 2025

6.15.2.1.24 Email Address

Location: /definitions/patents/[]/patentAssignee/[]/email
Property: email (Optional)
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)
```
firstname.lastname@example.com
```

6.15.2.1.25 Phone

Location: /definitions/patents/[]/patentAssignee/[]/phone
Property: phone (Optional)
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.15.2.1.26 BOM Reference

Location: /definitions/patents/[]/patentAssignee/[]/bom-ref
Property: bom-ref (Optional)
Type: String

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

6.15.2.1.27 Organization Name

Location: /definitions/patents/[]/patentAssignee/[]/name
Property: name (Optional)
Type: String

The name of the organization.

Example (Informative)
```
Example Inc.
```

6.15.2.1.28 Organization Address

Location: /definitions/patents/[]/patentAssignee/[]/address
Property: address (Optional)
Type: Object

The physical address (location) of the organization.

© Ecma International 2025

Table 491 — Properties for the address object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the address elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.15.2.1.29 BOM Reference

Location: /definitions/patents/[]/patentAssignee/[]/address/bom-ref
Property: bom-ref (Optional)
Type: String

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

6.15.2.1.30 Country

Location: /definitions/patents/[]/patentAssignee/[]/address/country
Property: country (Optional)
Type: String

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

6.15.2.1.31 Region

Location: /definitions/patents/[]/patentAssignee/[]/address/region
Property: region (Optional)
Type: String

The region or state in the country.

Example (Informative)

```
Texas
```

© Ecma International 2025

6.15.2.1.32 Locality

Location: /definitions/patents/[]/patentAssignee/[]/address/locality
Property: locality (Optional)
Type: String

The locality or city within the country.

Example (Informative)
```
Austin
```

6.15.2.1.33 Post Office Box Number

Location: /definitions/patents/[]/patentAssignee/[]/address/postOfficeBoxNumber
Property: postOfficeBoxNumber (Optional)
Type: String

The post office box number.

Example (Informative)
```
901
```

6.15.2.1.34 Postal Code

Location: /definitions/patents/[]/patentAssignee/[]/address/postalCode
Property: postalCode (Optional)
Type: String

The postal code.

Example (Informative)
```
78758
```

6.15.2.1.35 Street Address

Location: /definitions/patents/[]/patentAssignee/[]/address/streetAddress
Property: streetAddress (Optional)
Type: String

The street address.

Example (Informative)
```
100 Main Street
```

6.15.2.1.36 Organization URL(s)

Location: /definitions/patents/[]/patentAssignee/[]/url
Property: url (Optional)
Type: Array (of String)
Format: iri-reference as specified in RFC 3987

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

Example (Informative)
```
https://example.com
```


© Ecma International 2025

6.15.2.1.37 Organizational Contact

Location: /definitions/patents/[]/patentAssignee/[]/contact
Property: contact (Optional)
Type: Array

A contact at the organization. Multiple contacts are allowed. Each item of this array shall be an Organizational Person object.

6.15.2.1.38 Organizational Person

Location: /definitions/patents/[]/patentAssignee/[]/contact/[]
Type: Object

Table 492 — Properties for the contact object

Property	Type	Requirement	Description
bom-ref	String	Optional	An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall 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.

6.15.2.1.39 BOM Reference

Location: /definitions/patents/[]/patentAssignee/[]/contact/[]/bom-ref
Type: String

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

6.15.2.1.40 Name

Location: /definitions/patents/[]/patentAssignee/[]/contact/[]/name
Type: String

The name of a contact.

Example (Informative)

```
Contact name
```

6.15.2.1.41 Email Address

Location: /definitions/patents/[]/patentAssignee/[]/contact/[]/email
Type: String
Format: idn-email address as specified in RFC 6531

The email address of the contact.

Example (Informative)

```
firstname.lastname@example.com
```

© Ecma International 2025

6.15.2.1.42 Phone

Location: /definitions/patents/[]/patentAssignee/[]/contact/[]/phone
Type: String

The phone number of the contact.

Example (Informative)
```
800-555-1212
```

6.15.2.1.43 External References

Location: /definitions/patents/[]/externalReferences
Property: externalReferences (Optional)
Type: Array

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 shall be an External Reference object.

6.15.2.1.44 External Reference

Location: /definitions/patents/[]/externalReferences/[]
Type: Object

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 493 — Properties for the externalReference 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	A 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).
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.

© Ecma International 2025

6.15.2.1.45 URL

Location: /definitions/patents/[]/externalReferences/[]/url

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.

Shall be any of:

1. URL
2. BOM-Link

6.15.2.1.46 URL

Type: String
Format: iri-reference as specified in RFC 3987

6.15.2.1.47 BOM-Link

Type: Object

6.15.2.1.48 Comment

Location: /definitions/patents/[]/externalReferences/[]/comment
Type: String

A comment describing the external reference.

6.15.2.1.49 Type

Location: /definitions/patents/[]/externalReferences/[]/type
Type: String (enum)

Specifies the type of external reference.

Table 494 — 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.

© Ecma International 2025

Table 494 — Enumeration of possible values *(continued)*

Value	Description
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.
evidence	Information used to substantiate a claim.
formulation	Describes the formulation of any referencable object within the BOM, including components, services, metadata, declarations, or the BOM itself.
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 analyzes 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.

© Ecma International 2025

Table 494 — Enumeration of possible values *(continued)*

Value	Description
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.
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).
patent	References information about patents which may be defined in human-readable documents or in machine-readable formats such as CycloneDX or ST.96. For detailed patent information or to reference the information provided directly by patent offices, it is recommended to leverage standards from the World Intellectual Property Organization (WIPO) such as ST.96.
patent-family	References information about a patent family which may be defined in human-readable documents or in machine-readable formats such as CycloneDX or ST.96. A patent family is a group of related patent applications or granted patents that cover the same or similar invention. For detailed patent family information or to reference the information provided directly by patent offices, it is recommended to leverage standards from the World Intellectual Property Organization (WIPO) such as ST.96.

© Ecma International 2025

Table 494 — Enumeration of possible values *(continued)*

Value	Description
patent-assertion	References assertions made regarding patents associated with a component or service. Assertions distinguish between ownership, licensing, and other relevant interactions with patents.
citation	A reference to external citations applicable to the object identified by this BOM entry or the BOM itself. When used with a BOM-Link, this allows offloading citations into a separate CycloneDX BOM.
other	Use this if no other types accurately describe the purpose of the external reference.

6.15.2.1.50 Hashes

Location: /definitions/patents/[]/externalReferences/[]/hashes
Property: hashes (Optional)
Type: Array

The hashes of the external reference (if applicable). Each item of this array shall be a Hash object.

6.15.2.1.51 Hash

Location: /definitions/patents/[]/externalReferences/[]/hashes/[]
Type: Object

Table 495 — Properties for the hash object

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

6.15.2.1.52 Hash Algorithm

Location: /definitions/patents/[]/externalReferences/[]/hashes/[]/alg
Type: String (enum)

The algorithm that generated the hash value.

Enumeration of possible values:

- MD5
- SHA-1
- SHA-256
- SHA-384
- SHA-512
- SHA3-256
- SHA3-384
- SHA3-512
- BLAKE2b-256
- BLAKE2b-384
- BLAKE2b-512
- BLAKE3
- Streebog-256
- Streebog-512

© Ecma International 2025

6.15.2.1.53 Hash Value

Location: /definitions/patents/[]/externalReferences/[]/hashes/[]/content
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})$`

The value of the hash.

Example (Informative)
`3942447fac867ae5cdb3229b658f4d48`

6.15.2.1.54 Properties

Location: /definitions/patents/[]/externalReferences/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.15.2.1.55 Lightweight name-value pair

Location: /definitions/patents/[]/externalReferences/[]/properties/[]
Type: Object

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 496 — Properties for the property 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.

6.15.2.1.56 Name

Location: /definitions/patents/[]/externalReferences/[]/properties/[]/name
Type: String

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

6.15.2.1.57 Value

Location: /definitions/patents/[]/externalReferences/[]/properties/[]/value
Type: String

The value of the property.

© Ecma International 2025

6.15.2.1.58 BOM Reference

Location: /definitions/patents/[]/bom-ref
Property: bom-ref (Optional)
Type: String

An identifier which can be used to reference the object elsewhere in the BOM. Every **`bom-ref`** shall be unique within the BOM. For a patent, it might be a good idea to use a patent number as the BOM reference ID.

6.15.2.1.59 Patent Family ID

Location: /definitions/patents/[]/familyId
Property: familyId (Required)
Type: String

The unique identifier for the patent family, aligned with the **`id`** attribute in WIPO ST.96 v8.0's **`PatentFamilyType`**. Refer to PatentFamilyType in ST.96.

6.15.2.1.60 Priority Application

Location: /definitions/patents/[]/priorityApplication
Property: priorityApplication (Optional)
Type: Object

The priorityApplication contains the essential data necessary to identify and reference an earlier patent filing for priority rights. In line with WIPO ST.96 guidelines, it includes the jurisdiction (office code), application number, and filing date-the three key elements that uniquely specify the priority application in a global patent context.

Table 497 — Properties for the priorityApplication object

Property	Type	Requirement	Description
applicationNumber	String	Required	The unique number assigned to a patent application when it is filed with a patent office. It is used to identify the specific application and track its progress through the examination process. Aligned with **`ApplicationNumber`** in ST.96. Refer to ApplicationIdentificationType in ST.96.
jurisdiction	String	Required	The jurisdiction or patent office where the priority application was filed, specified using WIPO ST.3 codes. Aligned with **`IPOfficeCode`** in ST.96. Refer to IPOfficeCode in ST.96.
filingDate	String	Required	The date the priority application was filed, aligned with **`FilingDate`** in ST.96. Refer to FilingDate in ST.96.

6.15.2.1.61 Patent Application Number

Location: /definitions/patents/[]/priorityApplication/applicationNumber
Property: applicationNumber (Required)
Type: String
Pattern Constraint: `^[A-Za-z0-9][A-Za-z0-9\-/.()\s]{0,28}[A-Za-z0-9]$`

The unique number assigned to a patent application when it is filed with a patent office. It is used to identify the specific application and track its progress through the examination process. Aligned with **`ApplicationNumber`** in ST.96. Refer to ApplicationIdentificationType in ST.96.

© Ecma International 2025

Example 1 (Informative)
```
US20240000123
```

Example 2 (Informative)
```
EP23123456
```

6.15.2.1.62 Jurisdiction

Location: /definitions/patents/[]/priorityApplication/jurisdiction
Property: jurisdiction (Required)
Type: String
Pattern Constraint: `^[A-Z]{2}$`

The jurisdiction or patent office where the priority application was filed, specified using WIPO ST.3 codes. Aligned with **`IPOfficeCode`** in ST.96. Refer to IPOfficeCode in ST.96.

Example 1 (Informative)
```
US
```

Example 2 (Informative)
```
EP
```

Example 3 (Informative)
```
JP
```

6.15.2.1.63 Filing Date

Location: /definitions/patents/[]/priorityApplication/filingDate
Property: filingDate (Required)
Type: String
Format: date as specified in RFC 3339 section 5.6

The date the priority application was filed, aligned with **`FilingDate`** in ST.96. Refer to FilingDate in ST.96.

6.15.2.1.64 Family Members

Location: /definitions/patents/[]/members
Property: members (Optional)
Type: Array (of String)

A collection of patents or applications that belong to this family, each identified by a **`bom-ref`** pointing to a patent object defined elsewhere in the BOM. A **`bom-ref`** linking to a patent or application object within the BOM. Each item of this array shall be a string.

6.15.2.1.65 External References

Location: /definitions/patents/[]/externalReferences
Property: externalReferences (Optional)
Type: Array

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 shall be an External Reference object.

© Ecma International 2025

6.15.2.1.66 External Reference

Location: /definitions/patents/[]/externalReferences/[]
Type: Object

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 498 — Properties for the externalReference 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	A 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).
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.

6.15.2.1.67 URL

Location: /definitions/patents/[]/externalReferences/[]/url

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.

Shall be any of:

1. URL
2. BOM-Link

6.15.2.1.68 URL

Type: String
Format: iri-reference as specified in RFC 3987

6.15.2.1.69 BOM-Link

Type: Object

© Ecma International 2025

6.15.2.1.70 Comment

Location: /definitions/patents/[]/externalReferences/[]/comment
Type: String

A comment describing the external reference.

6.15.2.1.71 Type

Location: /definitions/patents/[]/externalReferences/[]/type
Type: String (enum)

Specifies the type of external reference.

Table 499 — 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.

© Ecma International 2025

Table 499 — Enumeration of possible values *(continued)*

Value	Description
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.
evidence	Information used to substantiate a claim.
formulation	Describes the formulation of any referencable object within the BOM, including components, services, metadata, declarations, or the BOM itself.
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 analyzes 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).

© Ecma International 2025

Table 499 — Enumeration of possible values *(continued)*

Value	Description
quality-metrics	Report or system in which quality metrics can be obtained.
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).
patent	References information about patents which may be defined in human-readable documents or in machine-readable formats such as CycloneDX or ST.96. For detailed patent information or to reference the information provided directly by patent offices, it is recommended to leverage standards from the World Intellectual Property Organization (WIPO) such as ST.96.
patent-family	References information about a patent family which may be defined in human-readable documents or in machine-readable formats such as CycloneDX or ST.96. A patent family is a group of related patent applications or granted patents that cover the same or similar invention. For detailed patent family information or to reference the information provided directly by patent offices, it is recommended to leverage standards from the World Intellectual Property Organization (WIPO) such as ST.96.
patent-assertion	References assertions made regarding patents associated with a component or service. Assertions distinguish between ownership, licensing, and other relevant interactions with patents.
citation	A reference to external citations applicable to the object identified by this BOM entry or the BOM itself. When used with a BOM-Link, this allows offloading citations into a separate CycloneDX BOM.
other	Use this if no other types accurately describe the purpose of the external reference.

6.15.2.1.72 Hashes

Location: /definitions/patents/[]/externalReferences/[]/hashes
Property: hashes (Optional)
Type: Array

The hashes of the external reference (if applicable). Each item of this array shall be a Hash object.

6.15.2.1.73 Hash

Location: /definitions/patents/[]/externalReferences/[]/hashes/[]
Type: Object

© Ecma International 2025

Table 500 — Properties for the hash object

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

6.15.2.1.74 Hash Algorithm

Location: /definitions/patents/[]/externalReferences/[]/hashes/[]/alg
Type: String (enum)

The algorithm that generated the hash value.

Enumeration of possible values:

- MD5
- SHA-1
- SHA-256
- SHA-384
- SHA-512
- SHA3-256
- SHA3-384
- SHA3-512
- BLAKE2b-256
- BLAKE2b-384
- BLAKE2b-512
- BLAKE3
- Streebog-256
- Streebog-512

6.15.2.1.75 Hash Value

Location: /definitions/patents/[]/externalReferences/[]/hashes/[]/content
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})$`

The value of the hash.

Example (Informative)
`3942447fac867ae5cdb3229b658f4d48`

6.15.2.1.76 Properties

Location: /definitions/patents/[]/externalReferences/[]/properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

© Ecma International 2025

6.15.2.1.77 Lightweight name-value pair

Location: /definitions/patents/[]/externalReferences/[]/properties/[]
Type: Object

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 501 — Properties for the property 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.

6.15.2.1.78 Name

Location: /definitions/patents/[]/externalReferences/[]/properties/[]/name
Type: String

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

6.15.2.1.79 Value

Location: /definitions/patents/[]/externalReferences/[]/properties/[]/value
Type: String

The value of the property.

6.16 Citations

Location: /citations
Property: citations (Optional)
Type: Array
Uniqueness: All items shall be unique.

A collection of attributions indicating which entity supplied information for specific fields within the BOM. Each item of this array shall be a Citation object.

6.16.1 Citation

Location: /citations/[]
Type: Object

Details a specific attribution of data within the BOM to a contributing entity or process.

© Ecma International 2025

Table 502 — Properties for the citation 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.
pointers	Array	Optional	One or more JSON Pointers identifying the BOM fields to which the attribution applies. Exactly one of the "pointers" or "expressions" elements shall be present.
expressions	Array	Optional	One or more path expressions used to locate values within a BOM. Exactly one of the "pointers" or "expressions" elements shall be present.
timestamp	String	Required	The date and time when the attribution was made or the information was supplied.
attributedTo	String	Optional	The **`bom-ref`** of an object, such as a component, service, tool, organisational entity, or person that supplied the cited information. At least one of the "attributedTo" or "process" elements shall be present.
process	String	Optional	The **`bom-ref`** to a process (such as a formula, workflow, task, or step) defined in the **`formulation`** section that executed or generated the attributed data. At least one of the "attributedTo" or "process" elements shall be present.
note	String	Optional	A description or comment about the context or quality of the data attribution.
signature	Array	Optional	A digital signature verifying the authenticity or integrity of the attribution.

6.16.1.1 BOM Reference

Location: /citations/[]/bom-ref
Type: String

Identifier for referable and therefore interlinkable elements. Value should not start with the BOM-Link intro 'urn:cdx:' to avoid conflicts with BOM-Links.

6.16.1.2 Field References

Location: /citations/[]/pointers
Property: pointers (Optional)
Type: Array (of String)

One or more JSON Pointers identifying the BOM fields to which the attribution applies. Exactly one of the "pointers" or "expressions" elements shall be present. A JSON Pointer identifying the BOM field to which the attribution applies. Users of other serialization formats (e.g. XML) shall use the JSON Pointer format to ensure consistent field referencing across representations. Each item of this array shall be a string.

6.16.1.3 Path Expressions

Location: /citations/[]/expressions
Property: expressions (Optional)
Type: Array (of String)

One or more path expressions used to locate values within a BOM. Exactly one of the "pointers" or "expressions" elements shall be present. Specifies a path expression used to locate a value within a BOM. The expression

© Ecma International 2025

syntax shall conform to the format of the BOM's serialization. Use JSONPath for JSON, XPath for XML, and default to JSONPath for Protocol Buffers unless otherwise specified. Implementers shall ensure the expression is valid within the context of the applicable serialization format. Each item of this array shall be a string.

6.16.1.4 Timestamp

Location: /citations/[]/timestamp
Type: String
Format: data-time as specified in RFC 3339 section 5.6

The date and time when the attribution was made or the information was supplied.

6.16.1.5 Attributed To

Location: /citations/[]/attributedTo
Type: String

The **`bom-ref`** of an object, such as a component, service, tool, organisational entity, or person that supplied the cited information. At least one of the "attributedTo" or "process" elements shall be present.

6.16.1.6 Process Reference

Location: /citations/[]/process
Type: String

The **`bom-ref`** to a process (such as a formula, workflow, task, or step) defined in the **`formulation`** section that executed or generated the attributed data. At least one of the "attributedTo" or "process" elements shall be present.

6.16.1.7 Note

Location: /citations/[]/note
Type: String

A description or comment about the context or quality of the data attribution.

6.16.1.8 Signature

Location: /citations/[]/signature
Type: Object

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.

© Ecma International 2025

6.17 Properties

Location: /properties
Property: properties (Optional)
Type: Array

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 shall be a Lightweight name-value pair object.

6.17.1 Lightweight name-value pair

Location: /properties/[]
Type: Object

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 503 — Properties for the property 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.

6.17.1.1 Name

Location: /properties/[]/name
Type: String

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

6.17.1.2 Value

Location: /properties/[]/value
Type: String

The value of the property.

6.18 Signature

Location: /signature
Property: signature (Optional)
Type: Object

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.

© Ecma International 2025

Bibliography

1. ISO-IEC 19770-2 *Software Identification (SWID) Tags*
   https://www.iso.org/standard/65666.html
2. IETF RFC 2045, *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*
   https://datatracker.ietf.org/doc/html/rfc2045
3. IETF RFC 6531, *SMTP Extension for Internationalized Email*
   https://datatracker.ietf.org/doc/html/rfc6531
4. IETF RFC 6901, *JavaScript Object Notation (JSON) Pointer*
   https://datatracker.ietf.org/doc/html/rfc6901
5. IETF RFC 7230, *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*
   https://datatracker.ietf.org/doc/html/rfc7230
6. IETF RFC 7296, *Internet Key Exchange Protocol Version 2*
   https://datatracker.ietf.org/doc/html/rfc7296
7. IETF RFC 9116, *A File Format to Aid in Security Vulnerability Disclosure*
   https://datatracker.ietf.org/doc/html/rfc9116
8. IETF RFC 9370, *Multiple Key Exchanges in the Internet Key Exchange Protocol Version 2*
   https://datatracker.ietf.org/doc/html/rfc9370
9. IETF RFC 9593, *Supported Authentication Methods in the Internet Key Exchange Protocol Version 2*
   https://datatracker.ietf.org/doc/html/rfc9593
10. IANA, *Media Types registry*
    https://www.iana.org/assignments/media-types/media-types.xhtml
11. IANA, *Gitoid scheme*
    https://www.iana.org/assignments/uri-schemes/prov/gitoid
12. OWASP, *CycloneDX Property Taxonomy*
    https://github.com/CycloneDX/cyclonedx-property-taxonomy
13. NIST, *CPE specification*
    https://nvd.nist.gov/products/cpe
14. Software Heritage Foundation, *SoftWare Heritage persistent IDentifiers (SWHIDs)*
    https://docs.softwareheritage.org/devel/swh-model/persistent-identifiers.html
15. World Intellectual Property Organization (WIPO) ST.96, *Processing of Intellectual Property information using XML*
    https://www.wipo.int/standards/en/st96/v9-0/

© Ecma International 2025

Colophon

This specification is authored on GitHub in a plaintext source format called Ecmarkup. Ecmarkup is an HTML and Markdown dialect that provides a framework and toolset for authoring ECMA specifications in plaintext and processing the specification into a full-featured HTML rendering that follows the editorial conventions for this document. Ecmarkup builds on and integrates a number of other formats and technologies including Grammarkdown for defining syntax and Ecmarkdown for authoring algorithm steps. PDF renderings of this specification are produced using a print stylesheet which takes advantage of the CSS Paged Media specification and is converted using PrinceXML.

We extend our gratitude to TC39 for their exceptional work in developing Ecmarkup, which has greatly facilitated TC54's successful adoption of this tool for the preparation and maintenance of our technical specifications.

Prior editions of this specification were transformed using Word, automated editing of Open Office XML, and Pandoc.

© Ecma International 2025

Software License

Ecma International
Rue du Rhone 114
CH-1204 Geneva
Tel: +41 22 849 6000
Fax: +41 22 849 6001
Web: https://ecma-international.org/

All Software contained in this document ("Software") is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT https://ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

© Ecma International 2025