BS ISO/IEC 20153:2025

BSI Standards Publication

Information technology — OASIS Common Security Advisory Framework (CSAF) v2.0 Specification

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 20153:2025

National foreword

This British Standard is the UK implementation of ISO/IEC 20153:2025.

The UK participation in its preparation was entrusted to Technical Committee ICT/1, Information systems co-ordination.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2025
Published by BSI Standards Limited 2025

ISBN 978 0 539 32548 5

ICS 35.020; 35.030

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 28 February 2025.

Amendments/corrigenda issued since publication

Date	Text affected

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 20153:2025

International Standard

ISO/IEC 20153

Information technology — OASIS Common Security Advisory Framework (CSAF) v2.0 Specification

First edition
2025-02

Reference number
ISO/IEC 20153:2025(en)

© ISO/IEC 2025

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 20153:2025

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2025

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

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

Published in Switzerland

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 20153:2025

Foreword

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

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

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

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

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

This document was prepared by OASIS [as OASIS Common Security Advisory Framework (CSAF) TC] and drafted in accordance with its editorial rules. It was adopted, under the JTC 1 PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

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

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

Table of Contents

- 1 Introduction
  - 1.1 IPR Policy
  - 1.2 Terminology
  - 1.3 Normative References
  - 1.4 Informative References
  - 1.5 Typographical Conventions
- 2 Design Considerations
  - 2.1 Construction Principles
- 3 Schema Elements
  - 3.1 Definitions
    - 3.1.1 Acknowledgments Type
      - 3.1.1.1 Acknowledgments Type - Names
      - 3.1.1.2 Acknowledgments Type - Organization
      - 3.1.1.3 Acknowledgments Type - Summary
      - 3.1.1.4 Acknowledgments Type - URLs
      - 3.1.1.5 Acknowledgments Type - Example
    - 3.1.2 Branches Type
      - 3.1.2.1 Branches Type - Branches
      - 3.1.2.2 Branches Type - Category
      - 3.1.2.3 Branches Type - Name
        - 3.1.2.3.1 Branches Type - Name under Product Version
        - 3.1.2.3.2 Branches Type - Name under Product Version Range
      - 3.1.2.4 Branches Type - Product
    - 3.1.3 Full Product Name Type
      - 3.1.3.1 Full Product Name Type - Name
      - 3.1.3.2 Full Product Name Type - Product ID
      - 3.1.3.3 Full Product Name Type - Product Identification Helper
        - 3.1.3.3.1 Full Product Name Type - Product Identification Helper - CPE
        - 3.1.3.3.2 Full Product Name Type - Product Identification Helper - Hashes
        - 3.1.3.3.3 Full Product Name Type - Product Identification Helper - Model Numbers
        - 3.1.3.3.4 Full Product Name Type - Product Identification Helper - PURL
        - 3.1.3.3.5 Full Product Name Type - Product Identification Helper - SBOM URLs
        - 3.1.3.3.6 Full Product Name Type - Product Identification Helper - Serial Numbers
        - 3.1.3.3.7 Full Product Name Type - Product Identification Helper - SKUs
        - 3.1.3.3.8 Full Product Name Type - Product Identification Helper - Generic URIs
    - 3.1.4 Language Type
    - 3.1.5 Notes Type
    - 3.1.6 Product Group ID Type
    - 3.1.7 Product Groups Type
    - 3.1.8 Product ID Type
    - 3.1.9 Products Type
    - 3.1.10 References Type
    - 3.1.11 Version Type
      - 3.1.11.1 Version Type - Integer versioning
      - 3.1.11.2 Version Type - Semantic versioning
  - 3.2 Properties
    - 3.2.1 Document Property
      - 3.2.1.1 Document Property - Acknowledgments
      - 3.2.1.2 Document Property - Aggregate Severity
      - 3.2.1.3 Document Property - Category
      - 3.2.1.4 Document Property - CSAF Version
      - 3.2.1.5 Document Property - Distribution
        - 3.2.1.5.1 Document Property - Distribution - Text
        - 3.2.1.5.2 Document Property - Distribution - TLP
      - 3.2.1.6 Document Property - Language

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

3.2.1.7 Document Property - Notes
3.2.1.8 Document Property - Publisher
3.2.1.8.1 Document Property - Publisher - Category
3.2.1.8.2 Document Property - Publisher - Contact Details
3.2.1.8.3 Document Property - Publisher - Issuing Authority
3.2.1.8.4 Document Property - Publisher - Name
3.2.1.8.5 Document Property - Publisher - Namespace
3.2.1.9 Document Property - References
3.2.1.10 Document Property - Source Language
3.2.1.11 Document Property - Title
3.2.1.12 Document Property - Tracking
3.2.1.12.1 Document Property - Tracking - Aliases
3.2.1.12.2 Document Property - Tracking - Current Release Date
3.2.1.12.3 Document Property - Tracking - Generator
3.2.1.12.4 Document Property - Tracking - ID
3.2.1.12.5 Document Property - Tracking - Initial Release Date
3.2.1.12.6 Document Property - Tracking - Revision History
3.2.1.12.7 Document Property - Tracking - Status
3.2.1.12.8 Document Property - Tracking - Version
3.2.2 Product Tree Property
3.2.2.1 Product Tree Property - Branches
3.2.2.2 Product Tree Property - Full Product Names
3.2.2.3 Product Tree Property - Product Groups
3.2.2.4 Product Tree Property - Relationships
3.2.3 Vulnerabilities Property
3.2.3.1 Vulnerabilities Property - Acknowledgments
3.2.3.2 Vulnerabilities Property - CVE
3.2.3.3 Vulnerabilities Property - CWE
3.2.3.4 Vulnerabilities Property - Discovery Date
3.2.3.5 Vulnerabilities Property - Flags
3.2.3.6 Vulnerabilities Property - IDs
3.2.3.7 Vulnerabilities Property - Involvements
3.2.3.8 Vulnerabilities Property - Notes
3.2.3.9 Vulnerabilities Property - Product Status
3.2.3.10 Vulnerabilities Property - References
3.2.3.11 Vulnerabilities Property - Release Date
3.2.3.12 Vulnerabilities Property - Remediations
3.2.3.12.1 Vulnerabilities Property - Remediations - Category
3.2.3.12.2 Vulnerabilities Property - Remediations - Date
3.2.3.12.3 Vulnerabilities Property - Remediations - Details
3.2.3.12.4 Vulnerabilities Property - Remediations - Entitlements
3.2.3.12.5 Vulnerabilities Property - Remediations - Group IDs
3.2.3.12.6 Vulnerabilities Property - Remediations - Product IDs
3.2.3.12.7 Vulnerabilities Property - Remediations - Restart Required
3.2.3.12.8 Vulnerabilities Property - Remediations - URL
3.2.3.13 Vulnerabilities Property - Scores
3.2.3.14 Vulnerabilities Property - Threats
3.2.3.15 Vulnerabilities Property - Title
4 Profiles
4.1 Profile 1: CSAF Base
4.2 Profile 2: Security incident response
4.3 Profile 3: Informational Advisory
4.4 Profile 4: Security Advisory
4.5 Profile 5: VEX
5 Additional Conventions
5.1 Filename
5.2 Separation in Data Stream
5.3 Sorting

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

6 Tests
6.1 Mandatory Tests
6.1.1 Missing Definition of Product ID
6.1.2 Multiple Definition of Product ID
6.1.3 Circular Definition of Product ID
6.1.4 Missing Definition of Product Group ID
6.1.5 Multiple Definition of Product Group ID
6.1.6 Contradicting Product Status
6.1.7 Multiple Scores with same Version per Product
6.1.8 Invalid CVSS
6.1.9 Invalid CVSS computation
6.1.10 Inconsistent CVSS
6.1.11 CWE
6.1.12 Language
6.1.13 PURL
6.1.14 Sorted Revision History
6.1.15 Translator
6.1.16 Latest Document Version
6.1.17 Document Status Draft
6.1.18 Released Revision History
6.1.19 Revision History Entries for Pre-release Versions
6.1.20 Non-draft Document Version
6.1.21 Missing Item in Revision History
6.1.22 Multiple Definition in Revision History
6.1.23 Multiple Use of Same CVE
6.1.24 Multiple Definition in Involvements
6.1.25 Multiple Use of Same Hash Algorithm
6.1.26 Prohibited Document Category Name
6.1.27 Profile Tests
6.1.27.1 Document Notes
6.1.27.2 Document References
6.1.27.3 Vulnerabilities
6.1.27.4 Product Tree
6.1.27.5 Vulnerability Notes
6.1.27.6 Product Status
6.1.27.7 VEX Product Status
6.1.27.8 Vulnerability ID
6.1.27.9 Impact Statement
6.1.27.10 Action Statement
6.1.27.11 Vulnerabilities
6.1.28 Translation
6.1.29 Remediation without Product Reference
6.1.30 Mixed Integer and Semantic Versioning
6.1.31 Version Range in Product Version
6.1.32 Flag without Product Reference
6.1.33 Multiple Flags with VEX Justification Codes per Product
6.2 Optional Tests
6.2.1 Unused Definition of Product ID
6.2.2 Missing Remediation
6.2.3 Missing Score
6.2.4 Build Metadata in Revision History
6.2.5 Older Initial Release Date than Revision History
6.2.6 Older Current Release Date than Revision History
6.2.7 Missing Date in Involvements
6.2.8 Use of MD5 as the only Hash Algorithm
6.2.9 Use of SHA-1 as the only Hash Algorithm
6.2.10 Missing TLP label
6.2.11 Missing Canonical URL

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

6.2.12 Missing Document Language
6.2.13 Sorting
6.2.14 Use of Private Language
6.2.15 Use of Default Language
6.2.16 Missing Product Identification Helper
6.2.17 CVE in field IDs
6.2.18 Product Version Range without vers
6.2.19 CVSS for Fixed Products
6.2.20 Additional Properties
6.3 Informative Test
6.3.1 Use of CVSS v2 as the only Scoring System
6.3.2 Use of CVSS v3.0
6.3.3 Missing CVE
6.3.4 Missing CWE
6.3.5 Use of Short Hash
6.3.6 Use of non-self referencing URLs Failing to Resolve
6.3.7 Use of self referencing URLs Failing to Resolve
6.3.8 Spell check
6.3.9 Branch Categories
6.3.10 Usage of Product Version Range
6.3.11 Usage of V as Version Indicator
7 Distributing CSAF documents
7.1 Requirements
7.1.1 Requirement 1: Valid CSAF document
7.1.2 Requirement 2: Filename
7.1.3 Requirement 3: TLS
7.1.4 Requirement 4: TLP:WHITE
7.1.5 Requirement 5: TLP:AMBER and TLP:RED
7.1.6 Requirement 6: No Redirects
7.1.7 Requirement 7: provider-metadata.json
7.1.8 Requirement 8: security.txt
7.1.9 Requirement 9: Well-known URL for provider-metadata.json
7.1.10 Requirement 10: DNS path
7.1.11 Requirement 11: One folder per year
7.1.12 Requirement 12: index.txt
7.1.13 Requirement 13: changes.csv
7.1.14 Requirement 14: Directory listings
7.1.15 Requirement 15: ROLIE feed
7.1.16 Requirement 16: ROLIE service document
7.1.17 Requirement 17: ROLIE category document
7.1.18 Requirement 18: Integrity
7.1.19 Requirement 19: Signatures
7.1.20 Requirement 20: Public OpenPGP Key
7.1.21 Requirement 21: List of CSAF providers
7.1.22 Requirement 22: Two disjoint issuing parties
7.1.23 Requirement 23: Mirror
7.2 Roles
7.2.1 Role: CSAF publisher
7.2.2 Role: CSAF provider
7.2.3 Role: CSAF trusted provider
7.2.4 Role: CSAF lister
7.2.5 Role: CSAF aggregator
7.3 Retrieving rules
7.3.1 Finding provider-metadata.json
7.3.2 Retrieving CSAF documents
8 Safety, Security, and Data Protection Considerations
9 Conformance
9.1 Conformance Targets

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

9.1.1 Conformance Clause 1: CSAF document
9.1.2 Conformance Clause 2: CSAF producer
9.1.3 Conformance Clause 3: CSAF direct producer
9.1.4 Conformance Clause 4: CSAF converter
9.1.5 Conformance Clause 5: CVRF CSAF converter
9.1.6 Conformance Clause 6: CSAF content management system
9.1.7 Conformance Clause 7: CSAF post-processor
9.1.8 Conformance Clause 8: CSAF modifier
9.1.9 Conformance Clause 9: CSAF translator
9.1.10 Conformance Clause 10: CSAF consumer
9.1.11 Conformance Clause 11: CSAF viewer
9.1.12 Conformance Clause 12: CSAF management system
9.1.13 Conformance Clause 13: CSAF asset matching system
9.1.14 Conformance Clause 14: CSAF basic validator
9.1.15 Conformance Clause 15: CSAF extended validator
9.1.16 Conformance Clause 16: CSAF full validator
9.1.17 Conformance Clause 17: CSAF SBOM matching system
Appendix A. Acknowledgments
Appendix B. Revision History
Appendix C. Guidance on the Size of CSAF Documents
C.1 File size
C.2 Array length
C.3 String length
C.4 URI length
C.5 Enum
C.6 Date

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

1 Introduction

1.1 IPR Policy

This specification is provided under the Non-Assertion Mode of the OASIS IPR Policy, the mode chosen when the Technical Committee was established. For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the TC's web page (https://www.oasis-open.org/committees/csaf/ipr.php).

1.2 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] and [RFC8174] when, and only when, they appear in all capitals, as shown here.

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

advisory: reporting item that describes a condition present in an artifact and that requires action by the consumers

advisory document: artifact in which an analysis tool reports a result

advisory management system: software system that consumes the documents produced by analysis tools, produces advisories that enable engineering and operating organizations to assess the quality of these software artifacts at a point in time, and performs functions such as filing security advisories and displaying information about individual advisories. **Note**: An advisory management system can interact with a document viewer to display information about individual advisories.

advisory matching: process of determining whether two advisories are targeting the same products and conditions

artifact: sequence of bytes addressable via a URI. *Examples*: A physical file in a file system such as a source file, an object file, a configuration file or a data file; a specific version of a file in a version control system; a database table accessed via an HTTP request; an arbitrary stream of bytes returned from an HTTP request, a product URL, a common product enumeration value.

CSAF asset matching system: program that connects to or is an asset database and is able to manage CSAF documents as required by CSAF management system as well as matching them to assets of the asset database.

CSAF basic validator: A program that reads a document and checks it against the JSON schema and performs mandatory tests.

CSAF consumer: program that reads and interprets a CSAF document

CSAF content management system: program that is able to create, review and manage CSAF documents and is able to preview their details as required by CSAF viewer.

CSAF converter: CSAF producer that transforms the output of an analysis tool from its native output format into the CSAF format

CSAF direct producer: analysis tool which acts as a CSAF producer

CSAF document: security advisory text document in the format defined by this document.

CSAF extended validator: A CSAF basic validator that additionally performs optional tests.

CSAF full validator: A CSAF extended validator that additionally performs informative tests.

CSAF management system: program that is able to manage CSAF documents and is able to display their details as required by CSAF viewer.

CSAF modifier: CSAF post-processor which takes a CSAF document as input and modifies the structure or values of properties. The output is a valid CSAF document.

CSAF post-processor: CSAF producer that transforms an existing CSAF document into a new CSAF document, for example, by removing or redacting elements according to sharing policies.

CSAF SBOM matching system: A program that connects to or is an SBOM database and is able to manage CSAF documents as required by CSAF management system as well as matching them to SBOM components of the SBOM database.

CSAF producer: program that emits output in the CSAF format

CSAF translator: CSAF post-processor which takes a CSAF document as input and translates values of properties into another language. The output is a valid CSAF document.

CSAF viewer: CSAF consumer that reads a CSAF document, displays a list of the results it contains, and allows an end user to view each

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

result in the context of the artifact in which it occurs.

CVRF CSAF converter: CSAF producer which takes a CVRF document as input and converts it into a valid CSAF document.

document: output file produced by an analysis tool, which enumerates the results produced by the tool

driver: tool component containing an analysis tool's or converter's primary executable, which controls the tool's or converter's execution, and which in the case of an analysis tool typically defines a set of analysis rules

embedded link: syntactic construct which enables a message string to refer to a location mentioned in the document

empty array: array that contains no elements, and so has a length of 0

empty object: object that contains no properties

empty string: string that contains no characters, and so has a length of 0

(end) user: person who uses the information in a document to investigate, triage, or resolve results

engineering system: software analysis environment within which analysis tools execute. **Note**: An engineering system might include a build system, a source control system, a result management system, a bug tracking system, a test execution system, and so on.

extension: tool component other than the driver (for example, a plugin, a configuration file, or a taxonomy)

external property file: file containing the values of one or more externalized properties

externalizable property: property that can be contained in an external property file

externalized property: property stored outside of the CSAF document to which it logically belongs

false positive: result which an end user decides does not actually represent a problem

fingerprint: stable value that can be used by a result management system to uniquely identify a result over time, even if a relevant artifact is modified

formatted message: message string which contains formatting information such as Markdown formatting characters

fully qualified logical name: string that fully identifies the programmatic construct specified by a logical location, typically by means of a hierarchical identifier.

hierarchical string: string in the format <component>{/<component>}*

line: contiguous sequence of characters, starting either at the beginning of an artifact or immediately after a newline sequence, and ending at and including the nearest subsequent newline sequence, if one is present, or else extending to the end of the artifact

line (number): 1-based index of a line within a file. **Note**: Abbreviated to "line" when there is no danger of ambiguity with "line" in the sense of a sequence of characters.

localizable: subject to being translated from one natural language to another

message string: human-readable string that conveys information relevant to an element in a CSAF document

nested artifact: artifact that is contained within another artifact

newline sequence: sequence of one or more characters representing the end of a line of text. **Note**: Some systems represent a newline sequence with a single newline character; others represent it as a carriage return character followed by a newline character.

notification: reporting item that describes a condition encountered by a tool during its execution

opaque: neither human-readable nor machine-parsable into constituent parts

parent (artifact): artifact which contains one or more nested artifacts

plain text message: message string which does not contain any formatting information

plugin: tool component that defines additional rules

policy: set of rule configurations that specify how results that violate the rules defined by a particular tool component are to be treated

problem: result which indicates a condition that has the potential to detract from the quality of the program. *Examples*: A security vulnerability, a deviation from contractual or legal requirements.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

product: is any deliverable (e.g. software, hardware, specification,...) which can be referred to with a name. This applies regardless of the origin, the license model, or the mode of distribution of the deliverable.

property: attribute of an object consisting of a name and a value associated with the name

redactable property: property that potentially contains sensitive information that a CSAF direct producer or a CSAF post-processor might wish to redact

reporting item: unit of output produced by a tool, either a result or a notification

reporting configuration: the subset of reporting metadata that a tool can configure at runtime, before performing its scan. *Examples*: severity level, rank

repository container for a related set of files in a version control system

taxonomy: classification of analysis results into a set of categories

tag: string that conveys additional information about the CSAF document element to which it applies

text artifact: artifact considered as a sequence of characters organized into lines and columns

text region: region representing a contiguous range of zero or more characters in a text artifact

tool component: component of an analysis tool or converter, either its driver or an extension, consisting of one or more files

top-level artifact: artifact which is not contained within any other artifact

translation: rendering of a tool component's localizable strings into another language

triage: decide whether a result indicates a problem that needs to be corrected

user: see end user.

VCS: version control system

vendor: the community, individual, or organization that created or maintains a product (including open source software and hardware providers)

VEX: Vulnerability Exploitability eXchange - enables a supplier or other party to assert whether or not a particular product is affected by a specific vulnerability, especially helpful in efficiently consuming SBOM data.

viewer: see CSAF viewer.

vulnerability: functional behavior of a product or service that violates an implicit or explicit security policy (conforming to ISO/IEC 29147 [ISO29147])

XML: eXtensible Markup Language - the format used by the predecessors of this standard, namely CVRF 1.1 and CVRF 1.2.

1.3 Normative References

[JSON-Schema-Core]

JSON Schema: A Media Type for Describing JSON Documents, draft-bhutton-json-schema-00, December 2020, https://datatracker.ietf.org/doc/html/draft-bhutton-json-schema-00.

[JSON-Schema-Validation]

JSON Schema Validation: A Vocabulary for Structural Validation of JSON, draft-bhutton-json-schema-validation-00, December 2020, https://datatracker.ietf.org/doc/html/draft-bhutton-json-schema-validation-00.

[JSON-Hyper-Schema]

JSON Hyper-Schema: A Vocabulary for Hypermedia Annotation of JSON, draft-handrews-json-schema-hyperschema-02, September 2019, https://json-schema.org/draft/2019-09/json-schema-hypermedia.html.

[Relative-JSON-Pointers]

Relative JSON Pointers, draft-bhutton-relative-json-pointer-00, December 2020, https://datatracker.ietf.org/doc/html/draft-bhutton-relative-json-pointer-00.

[RFC2119]

Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997,

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 20153:2025

https://www.rfc-editor.org/info/rfc2119.

[RFC7464]

Williams, N., "JavaScript Object Notation (JSON) Text Sequences", RFC 7464, DOI 10.17487/RFC7464, February 2015, https://www.rfc-editor.org/info/rfc7464.

[RFC8174]

Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, https://www.rfc-editor.org/info/rfc8174.

[RFC8259]

T. Bray, Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 8259, DOI 10.17487/RFC8259, December 2017, https://www.rfc-editor.org/info/rfc8259.

1.4 Informative References

[CPE23-A]

Common Platform Enumeration: Applicability Language Specification Version 2.3 (NISTIR 7698), D. Waltermire, P. Cichonski, K. Scarfone, Editors, NIST Interagency Report 7698, August 2011, https://dx.doi.org/10.6028/NIST.IR.7698.

[CPE23-D]

Common Platform Enumeration: Dictionary Specification Version 2.3, P. Cichonski, D. Waltermire, K. Scarfone, Editors, NIST Interagency Report 7697, August 2011, https://dx.doi.org/10.6028/NIST.IR.7697.

[CPE23-M]

Common Platform Enumeration: Naming Matching Specification Version 2.3, M. Parmelee, H. Booth, D. Waltermire, K. Scarfone, Editors, NIST Interagency Report 7696, August 2011, https://dx.doi.org/10.6028/NIST.IR.7696.

[CPE23-N]

Common Platform Enumeration: Naming Specification Version 2.3, B. Cheikes, D. Waltermire, K. Scarfone, Editors, NIST Interagency Report 7695, August 2011, https://dx.doi.org/10.6028/NIST.IR.7695.

[CVE]

Common Vulnerability and Exposures (CVE) – The Standard for Information Security Vulnerability Names, MITRE, 1999, https://cve.mitre.org/about/.

[CVE-NF]

Common Vulnerability and Exposures (CVE) – The Standard for Information Security Vulnerability Names - CVE ID Syntax Change, MITRE, January 01, 2014, https://cve.mitre.org/cve/identifiers/syntaxchange.html.

[CVRF-1-1]

The Common Vulnerability Reporting Framework (CVRF) Version 1.1, M. Schiffman, Editor, May 2012, Internet Consortium for Advancement of Security on the Internet (ICASI), https://www.icasi.org/the-common-vulnerability-reporting-framework-cvrf-v1-1/.

[CVRF-v1.2]

CSAF Common Vulnerability Reporting Framework (CVRF) Version 1.2. Edited by Stefan Hagen. 13 September 2017. OASIS Committee Specification 01. https://docs.oasis-open.org/csaf/csaf-cvrf/v1.2/cs01/csaf-cvrf-v1.2-cs01.html. Latest version: https://docs.oasis-open.org/csaf/csaf-cvrf/v1.2/csaf-cvrf-v1.2.html.

[CVSS2]

A Complete Guide to the Common Vulnerability Scoring System Version 2.0, P. Mell, K. Scarfone, S. Romanosky, Editors, First.org, Inc., June 2007, https://www.first.org/cvss/cvss-v2-guide.pdf.

[CVSS30]

Common Vulnerability Scoring System v3.0: Specification Document, FIRST.Org, Inc., June 2019, https://www.first.org/cvss/v3.0/cvss-v30-specification_v1.9.pdf.

[CVSS31]

Common Vulnerability Scoring System v3.1: Specification Document, FIRST.Org, Inc., June 2019, https://www.first.org/cvss/v3-1/cvss-v31-

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 20153:2025

specification_r1.pdf.

[CWE]

Common Weakness Enumeration (CWE) – A Community-Developed List of Software Weakness Types, MITRE, 2005, http://cwe.mitre.org/about/.

[CYCLONEDX13]

CycloneDX Software Bill-of-Material Specification JSON schema version 1.3, cyclonedx.org, May 2021, https://github.com/CycloneDX/specification/blob/1.3/schema/bom-1.3.schema.json.

[GFMCMARK]

GitHub's fork of cmark, a CommonMark parsing and rendering library and program in C, https://github.com/github/cmark.

[GFMENG]

GitHub Engineering: A formal spec for GitHub Flavored Markdown, https://githubengineering.com/a-formal-spec-for-github-markdown/.

[ISO8601]

Data elements and interchange formats — Information interchange — Representation of dates and times, International Standard, ISO 8601:2004(E), December 1, 2004, https://www.iso.org/standard/40874.html.

[ISO19770-2]

Information technology — IT asset management — Part 2: Software identification tag, International Standard, ISO 19770-2:2015, September 30, 2015, https://www.iso.org/standard/65666.html.

[ISO29147]

Information technology — Security techniques — Vulnerability disclosure, International Standard, ISO/IEC 29147:2018, October, 2018, https://www.iso.org/standard/72311.html.

[OPENSSL]

GTLS/SSL and crypto library, OpenSSL Software Foundation, https://www.openssl.org/.

[PURL]

Package URL (PURL), GitHub Project, https://github.com/package-url/purl-spec.

[RFC3339]

Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, https://www.rfc-editor.org/info/rfc3339.

[RFC3552]

Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, https://www.rfc-editor.org/info/rfc3552.

[RFC3986]

Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, https://www.rfc-editor.org/info/rfc3986.

[RFC4880]

Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, DOI 10.17487/RFC4880, November 2007, https://www.rfc-editor.org/info/rfc4880.

[RFC7231]

Fielding, R., Ed., and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, https://www.rfc-editor.org/info/rfc7231.

[RFC7464]

N. Williams., "JavaScript Object Notation (JSON) Text Sequences", RFC 7464, DOI 10.17487/RFC7464, February 2015, https://www.rfc-editor.org/info/rfc7464.

[RFC8615]

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 20153:2025

Nottingham, M., "Well-Known Uniform Resource Identifiers (URIs)", RFC 8615, DOI 10.17487/RFC8615, May 2019, https://www.rfc-editor.org/info/rfc8615.

[RFC9116]

Foudil, E. and Y. Shafranovich, "A File Format to Aid in Security Vulnerability Disclosure", RFC 9116, DOI 10.17487/RFC9116, April 2022, https://www.rfc-editor.org/info/rfc9116.

[SCAP12]

The Technical Specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.2, D. Waltermire, S. Quinn, K. Scarfone, A. Halbardier, Editors, NIST Spec. Publ. 800-126 rev. 2, September 2011, https://dx.doi.org/10.6028/NIST.SP.800-126r2.

[SECURITY-TXT]

Foudil, E. and Shafranovich, Y., *Security.txt Project*, https://securitytxt.org/.

[SemVer]

Semantic Versioning 2.0.0, T. Preston-Werner, June 2013, https://semver.org/.

[SPDX22]

The Software Package Data Exchange (SPDX®) Specification Version 2.2, Linux Foundation and its Contributors, 2020, https://spdx.github.io/spdx-spec/.

[VERS]

vers: a mostly universal version range specifier, Part of the PURL GitHub Project, https://github.com/package-url/purl-spec/blob/version-range-spec/VERSION-RANGE-SPEC.rst.

[VEX]

Vulnerability-Exploitability eXchange (VEX) - An Overview, VEX sub-group of the Framing Working Group in the NTIA SBOM initiative, 27 September 2021, https://ntia.gov/files/ntia/publications/vex_one-page_summary.pdf.

[VEX-Justification]

Vulnerability Exploitability eXchange (VEX) - Status Justifications, VEX sub-group of the Framing Working Group in the CISA SBOM initiative, XX May 2022, https://www.cisa.gov/sites/default/files/publications/VEX_Status_Justification_Jun22.pdf.

[XML]

Extensible Markup Language (XML) 1.0 (Fifth Edition), T. Bray, J. Paoli, M. Sperberg-McQueen, E. Maler, F. Yergeau, Editors, W3C Recommendation, November 26, 2008, https://www.w3.org/TR/2008/REC-xml-20081126/. Latest version available at https://www.w3.org/TR/xml.

[XML-Schema-1]

W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures, S. Gao, M. Sperberg-McQueen, H. Thompson, N. Mendelsohn, D. Beech, M. Maloney, Editors, W3C Recommendation, April 5, 2012, https://www.w3.org/TR/2012/REC-xmlschema11-1-20120405/. Latest version available at https://www.w3.org/TR/xmlschema11-1/.

[XML-Schema-2]

W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes, D. Peterson, S. Gao, A. Malhotra, M. Sperberg-McQueen, H. Thompson, Paul V. Biron, Editors, W3C Recommendation, April 5, 2012, https://www.w3.org/TR/2012/REC-xmlschema11-2-20120405/. Latest version available at https://www.w3.org/TR/xmlschema11-2/.

1.5 Typographical Conventions

Keywords defined by this specification use this `monospaced` font.

```
Normative source code uses this paragraph style.
```

Some sections of this specification are illustrated with non-normative examples introduced with "Example" or "Examples" like so:

Examples 4321:

```
Informative examples also use this paragraph style but preceded by the text "Example(s)".
```

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

All examples in this document are informative only.

All other text is normative unless otherwise labeled e.g. like the following informative comment:

> This is a pure informative comment that may be present, because the information conveyed is deemed useful advice or common pitfalls learned from implementer or operator experience and often given including the rationale.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

2 Design Considerations

The Common Security Advisory Framework (CSAF) is a language to exchange Security Advisories formulated in JSON.

The term Security Advisory as used in this document describes any notification of security issues in products of and by providers. Anyone providing a product is considered in this document as a vendor, i.e. developers or maintainers of information system products or services. This includes all authoritative product vendors, Product Security Incident Response Teams (PSIRTs), and product resellers and distributors, including authoritative vendor partners. A security issue is not necessarily constrained to a problem statement, the focus of the term is on the security aspect impacting (or not impacting) specific product-platform-version combinations. Information on presence or absence of workarounds is also considered part of the security issue. This document is the definitive reference for the language elements of CSAF version 2.0. The encompassing JSON schema file noted in the Additional Artifacts section of the title page SHALL be taken as normative in the case a gap or an inconsistency in this explanatory document becomes evident. The following presentation in this section is grouped by topical area, and is not simply derivative documentation from the schema document itself. The information contained aims to be more descriptive and complete. Where applicable, common conventions are stated and known common issues in usage are pointed out informatively to support implementers of document producers and consumers alike.

This minimal required information set does not provide any useful information on products, vulnerabilities, or security advisories. Thus, any real-world Security Advisory will carry additional information as specified in section 3 Schema elements.

Care has been taken, to design the containers for product and vulnerability information to support fine-grained mapping of security advisories onto product and vulnerability and minimize data duplication through referencing. The display of the elements representing Product Tree and Vulnerability information has been placed in the sections named accordingly.

2.1 Construction Principles

A Security Advisory defined as a CSAF document is the result of complex orchestration of many players and distinct and partially difficult to play schemas.

The format chosen is [JSONSchema] which allows validation and delegation to sub schema providers. The latter aligns well with separation of concerns and shares the format family of information interchange utilized by the providers of product and vulnerability information which migrated from XML to JSON since the creation of CSAF CVRF version 1.2, the predecessor of this specification.

The acronym CSAF, “Common Security Advisory Framework”, stands for the target of concerted mitigation and remediation accomplishment.

Technically, the use of JSON schema allows validation and proof of model conformance (through established schema based validation) of the declared information inside CSAF documents.

The CSAF schema structures its derived documents into three main classes of the information conveyed:

1. The frame, aggregation, and reference information of the document
2. Product information considered relevant by the creator
3. Vulnerability information and its relation to the products declared in 2.

Wherever possible repetition of data has been replaced by linkage through ID elements. Consistency on the content level thus is in the responsibility of the producer of such documents, to link e.g. vulnerability information to the matching product.

A dictionary like presentation of all defined schema elements is given in the section 3. Any expected relations to other elements (linkage) is described there. This linking relies on setting attribute values accordingly (mostly guided by industry best practice and conventions) and thus implies, that any deep validation on a semantic level (e.g. does the CWE match the described vulnerability) is to be ensured by the producer and consumer of CSAF documents. It is out of scope for this specification.

Proven and intended usage patterns from practice are given where possible.

Delegation to industry best practices technologies is used in referencing schemas for:

- Platform Data:
  - Common Platform Enumeration (CPE) Version 2.3 [CPE23-N]
- Vulnerability Scoring:
  - Common Vulnerability Scoring System (CVSS) Version 3.1 [CVSS31]
    - JSON Schema Reference https://www.first.org/cvss/cvss-v3.1.json
  - Common Vulnerability Scoring System (CVSS) Version 3.0 [CVSS30]
    - JSON Schema Reference https://www.first.org/cvss/cvss-v3.0.json
  - Common Vulnerability Scoring System (CVSS) Version 2.0 [CVSS2]
    - JSON Schema Reference https://www.first.org/cvss/cvss-v2.0.json
- Vulnerability Classification
  - Common Weakness Enumeration (CWE) [CWE]

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

        - CWE List: http://cwe.mitre.org/data/index.html
- Classification for Document Distribution
    - Traffic Light Protocol (TLP)
        - Default Definition: https://www.first.org/tlp/

Even though the JSON schema does not prohibit specifically additional properties and custom keywords, it is strongly recommended not to use them. Suggestions for new fields SHOULD be made through issues in the TC's GitHub.

> The standardized fields allow for scalability across different issuing parties and dramatically reduce the human effort and need for dedicated parsers as well as other tools on the side of the consuming parties.

Section 4 defined profiles that are used to ensure a common understanding of which fields are required in a given use case. Additional conventions are stated in section 5. The tests given in section 6 support CSAF producers and consumers to verify rules from the specification which can not be tested by the schema. Section 7 states how to distribute and where to find CSAF documents. Safety, Security and Data Protection are considered in section 8. Finally, a set of conformance targets describes tools in the ecosystem.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

3 Schema Elements

The CSAF schema describes how to represent security advisory information as a JSON document.

The CSAF schema Version 2.0 builds on the JSON Schema draft 2020-12 rules.

```
"$schema": "https://json-schema.org/draft/2020-12/schema"
```

The schema identifier is:

```
"$id": "https://docs.oasis-open.org/csaf/csaf/v2.0/csaf_json_schema.json"
```

The further documentation of the schema is organized via Definitions and Properties.

- Definitions provide types that extend the JSON schema model
- Properties use these types to support assembling security advisories

Types and properties together provide the vocabulary for the domain specific language supporting security advisories.

The single mandatory property is `document`. The optional two additional properties are `product_tree` and `vulnerabilities`.

3.1 Definitions

The definitions (`$defs`) introduce the following domain specific types into the CSAF language: Acknowledgments (`acknowledgments_t`), Branches (`branches_t`), Full Product Name (`full_product_name_t`), Language (`lang_t`), Notes (`notes_t`), Product Group ID (`product_group_id_t`), Product Groups (`product_groups_t`), Product ID (`product_id_t`), Products (`products_t`), References (`references_t`), and Version (`version_t`).

```
"$defs": {
    "acknowledgments_t": {
        // ...
    },
    "branches_t": {
        // ...
    },
    "full_product_name_t": {
        // ...
    },
    "lang_t": {
        // ...
    },
    "notes_t": {
        // ...
    },
    "product_group_id_t": {
        // ...
    },
    "product_groups_t": {
         // ...
    },
    "product_id_t": {
        // ...
    },
    "products_t": {
        // ...
    },
    "references_t": {
        // ...
    },
    "version_t": {
        // ...
    }
},
```

3.1.1 Acknowledgments Type

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

List of Acknowledgments (`acknowledgments_t`) type instances of value type `array` with 1 or more elements contain a list of `Acknowledgment` elements.

```
"acknowledgments_t": {
  // ...
  "items": {
    // ...
  }
},
```

The value type of Acknowledgment is `object` with at least 1 and at most 4 properties. Every such element acknowledges contributions by describing those that contributed. The properties are: `names`, `organization`, `summary`, and `urls`.

```
    "properties": {
      "names": {
        // ...
      },
      "organization": {
        // ...
      },
      "summary": {
        // ...
      },
      "urls": {
        // ...
      }
    }
```

3.1.1.1 Acknowledgments Type - Names

List of acknowledged names (`names`) has value type `array` with 1 or more items holds the names of contributors being recognized. Every such item of value type `string` with 1 or more characters represents the name of the contributor and contains the name of a single contributor being recognized.

Examples 1:

```
Albert Einstein
Johann Sebastian Bach
```

3.1.1.2 Acknowledgments Type - Organization

The contributing organization (`organization`) has value type `string` with 1 or more characters and holds the name of the contributing organization being recognized.

Examples 2:

```
CISA
Google Project Zero
Talos
```

3.1.1.3 Acknowledgments Type - Summary

Summary of the acknowledgment (`summary`) of value type `string` with 1 or more characters SHOULD represent any contextual details the document producers wish to make known about the acknowledgment or acknowledged parties.

Example 3:

```
First analysis of Coordinated Multi-Stream Attack (CMSA)
```

3.1.1.4 Acknowledgments Type - URLs

List of URLs (`urls`) of acknowledgment is a container (value type `array`) for 1 or more `string` of type URL that specifies a list of URLs or location of the reference to be acknowledged. Any URL of acknowledgment contains the URL or location of the reference to be acknowledged. Value type is string with format URI (`uri`).

3.1.1.5 Acknowledgments Type - Example

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

Example 4:

```
"acknowledgments": [
  {
    "names": [
      "Johann Sebastian Bach",
      "Georg Philipp Telemann",
      "Georg Friedrich Händel"
    ],
    "organization": "Baroque composers",
    "summary": "wonderful music"
  },
  {
    "organization": "CISA",
    "summary": "coordination efforts",
    "urls": [
      "https://cisa.gov"
    ]
  },
  {
    "organization": "BSI",
    "summary": "assistance in coordination"
  },
  {
    "names": [
      "Antonio Vivaldi"
    ],
    "summary": "influencing other composers"
  }
],
```

The example 4 above SHOULD lead to the following outcome in a human-readable advisory:

> We thank the following parties for their efforts:
>
> - Johann Sebastian Bach, Georg Philipp Telemann, Georg Friedrich Händel from Baroque composers for wonderful music
> - CISA for coordination efforts (see: https://cisa.gov)
> - BSI for assistance in coordination
> - Antonio Vivaldi for influencing other composers

3.1.2 Branches Type

List of branches (`branches_t`) with value type `array` contains 1 or more branch elements as children of the current element.

```
"branches_t": {
  //...
  "items": {
    // ...
  }
},
```

Every Branch holds exactly 3 properties and is a part of the hierarchical structure of the product tree. The properties `name` and `category` are mandatory. In addition, the object contains either a `branches` or a `product` property.

```
    "properties": {
      "branches": {
        // ...
      },
      "category": {
        // ...
      },
      "name": {
        // ...
      },
      "product": {
        // ...
```

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

```
        }
      }
```

> `branches_t` supports building a hierarchical structure of products that allows to indicate the relationship of products to each other and enables grouping for simpler referencing. As an example, the structure MAY use the following levels: `vendor` -> `product_family` -> `product_name` -> `product_version`. It is recommended to use the hierarchical structure of `vendor` -> `product_name` -> `product_version` whenever possible to support the identification and matching of products on the consumer side.

3.1.2.1 Branches Type - Branches

List of branches (`branches`) has the value type `branches_t`.

3.1.2.2 Branches Type - Category

Category of the branch (`category`) of value type `string` and `enum` describes the characteristics of the labeled branch. Valid `enum` values are:

```
architecture
host_name
language
legacy
patch_level
product_family
product_name
product_version
product_version_range
service_pack
specification
vendor
```

The value `architecture` indicates the architecture for which the product is intended.

The value `host_name` indicates the host name of a system/service.

The value `language` indicates the language of the product.

The value `legacy` indicates an entry that has reached its end of life.

The value `patch_level` indicates the patch level of the product.

The value `product_family` indicates the product family that the product falls into.

The value `product_name` indicates the name of the product.

The value `product_version` indicates exactly a single version of the product. The value of the adjacent `name` property can be numeric or some other descriptor. However, it MUST NOT contain version ranges of any kind.

> It is recommended to enumerate versions wherever possible. Nevertheless, the TC understands that this is sometimes impossible. To reflect that in the specification and aid in automatic processing of CSAF documents the value `product_version_range` was introduced. See next section for details.

The value `product_version_range` indicates a range of versions for the product. The value of the adjacent `name` property SHOULD NOT be used to convey a single version.

The value `service_pack` indicates the service pack of the product.

The value `specification` indicates the specification such as a standard, best common practice, etc.

The value `vendor` indicates the name of the vendor or manufacturer that makes the product.

3.1.2.3 Branches Type - Name

Name of the branch (`name`) of value type `string` with 1 or more characters contains the canonical descriptor or 'friendly name' of the branch.

Examples 5:

```
10
365
Microsoft
```

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

```
Office
PCS 7
SIMATIC
Siemens
Windows
```

A leading `v` or `V` in the value of `name` SHOULD only exist for the categories `product_version` or `product_version_range` if it is part of the product version as given by the vendor.

3.1.2.3.1 Branches Type - Name under Product Version

If adjacent property `category` has the value `product_version`, the value of `name` MUST NOT contain version ranges of any kind.

Examples 6 for `name` when using `product_version`:

```
10
17.4
v3
```

> The `product_version` is the easiest way for users to determine whether their version is meant (provided that the given ancestors in the product tree matched): If both version strings are the same, it is a match - otherwise not. Therefore, it is always recommended to enumerate product versions instead of providing version ranges.

Examples 7 for `name` when using `product_version` which are invalid:

```
8.0.0 - 8.0.1
8.1.5 and later
<= 2
prior to 4.2
All versions < V3.0.29
V3.0, V4.0, V4.1, V4.2
```

> All the examples above contain some kind of a version range and are therefore invalid under the category `product_version`.

3.1.2.3.2 Branches Type - Name under Product Version Range

If adjacent property `category` has the value `product_version_range`, the value of `name` MUST contain version ranges. The value of MUST obey to exactly one of the following options:

1. Version Range Specifier (vers)

   > vers is an ongoing community effort to address the problem of version ranges. Its draft specification is available at [VERS].

   vers MUST be used in its canonical form. To convey the term "all versions" the special string `vers:all/*` MUST be used.

   Examples 8 for `name` when using `product_version_range` with vers:

   ```
   vers:gem/>=2.2.0|!= 2.2.1|<2.3.0
   vers:npm/1.2.3|>=2.0.0|<5.0.0
   vers:pypi/0.0.0|0.0.1|0.0.2|0.0.3|1.0|2.0pre1
   vers:tomee/>=8.0.0-M1|<=8.0.1
   ```

   > Through the definitions of the vers specification a user can compute whether a given version is in a given range.

2. Vers-like Specifier (vls)

   This option uses only the `<version-constraint>` part from the vers specification. It MUST NOT have an URI nor the `<versioning-scheme>` part. It is a fallback option and SHOULD NOT be used unless really necessary.

   > The reason for that is, that it is nearly impossible for tools to reliable determine whether a given version is in the range or not.

   Tools MAY support this on best effort basis.

   Examples 9 for `name` when using `product_version_range` with vls:

   ```
   <=2
   ```

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

```
<4.2
<V3.0.29
>=8.1.5
```

3.1.2.4 Branches Type - Product

Product (`product`) has the value type Full Product Name (`full_product_name_t`).

3.1.3 Full Product Name Type

Full Product Name (`full_product_name_t`) with value type `object` specifies information about the product and assigns the product ID. The properties `name` and `product_id` are required. The property `product_identification_helper` is optional.

```
"full_product_name_t": {
  // ...
  "properties": {
    "name": {
      // ...
    },
    "product_id": {
      // ...
    },
    "product_identification_helper": {
      // ...
    }
  }
},
```

3.1.3.1 Full Product Name Type - Name

Textual description of the product (`name`) has value type `string` with 1 or more characters. The value SHOULD be the product's full canonical name, including version number and other attributes, as it would be used in a human-friendly document.

Examples 10:

```
Cisco AnyConnect Secure Mobility Client 2.3.185
Microsoft Host Integration Server 2006 Service Pack 1
```

3.1.3.2 Full Product Name Type - Product ID

Product ID (`product_id`) holds a value of type Product ID (`product_id_t`).

3.1.3.3 Full Product Name Type - Product Identification Helper

Helper to identify the product (`product_identification_helper`) of value type `object` provides in its properties at least one method which aids in identifying the product in an asset database. Of the given eight properties `cpe`, `hashes`, `model_numbers`, `purl`, `sbom_urls`, `serial_numbers`, `skus`, and `x_generic_uris`, one is mandatory.

```
"product_identification_helper": {
  // ...
  "properties": {
    "cpe": {
      // ...
    },
    "hashes": {
      // ...
    },
    "model_numbers": {
      // ...
    },
    "purl": {
      // ...
    },
    "sbom_urls": {
      // ...
    },
    "serial_numbers": {
```

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

```
          // ...
        },
        "skus": {
          // ...
        },
        "x_generic_uris": {
          // ...
        }
      }
```

3.1.3.3.1 Full Product Name Type - Product Identification Helper - CPE

Common Platform Enumeration representation (`cpe`) of value type `string` of 5 or more characters with `pattern` (regular expression):

```
    ^(cpe:2\\.3:[aho\\*\\-](:(((\\?*|\\*?)([a-zA-Z0-9\\-\\._]|(\\\\[\\\\\\*\\?!\"#\\$%&'\\(\\)\\+,/:;<=>@\\[\\]\\^`\\
{\\|\\}~]))+(\\?*|\\*?))|[\\*\\-]))){5}(:(([a-zA-Z]{2,3}(-([a-zA-Z]{2}|[0-9]{3}))?)|[\\*\\-]))(:(((\\?*|\\*?)([a-zA-Z0-
9\\-\\._]|(\\\\[\\\\\\*\\?!\"#\\$%&'\\(\\)\\+,/:;<=>@\\[\\]\\^`\\{\\|\\}~]))+(\\?*|\\*?))|[\\*\\-]))){4})|([c][pP]
[eE]:/[AHOaho]?(:[A-Za-z0-9\\._\\-~%]*){0,6})$
```

The Common Platform Enumeration (CPE) attribute refers to a method for naming platforms external to this specification. See [CPE23-N] for details.

3.1.3.3.2 Full Product Name Type - Product Identification Helper - Hashes

List of hashes (`hashes`) of value type `array` holding at least one item contains a list of cryptographic hashes usable to identify files.

```
    "hashes": {
      // ...
      "items": {
        // ...
      }
    },
```

Cryptographic hashes of value type `object` contains all information to identify a file based on its cryptographic hash values. Any cryptographic hashes object has the 2 mandatory properties `file_hashes` and `filename`.

```
        "properties": {
          "file_hashes": {
            // ...
          },
          "filename": {
            // ...
          }
        }
```

List of file hashes (`file_hashes`) of value type `array` holding at least one item contains a list of cryptographic hashes for this file.

```
    "file_hashes": {
      // ...
      "items": {
        // ...
      }
    },
```

Each File hash of value type `object` contains one hash value and algorithm of the file to be identified. Any File hash object has the 2 mandatory properties `algorithm` and `value`.

```
        "properties": {
          "algorithm": {
            // ...
          },
          "value": {
            // ...
          }
        }
```

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

The algorithm of the cryptographic hash representation (`algorithm`) of value type `string` with one or more characters contains the name of the cryptographic hash algorithm used to calculate the value. The default value for `algorithm` is `sha256`.

Examples 11:

```
blake2b512
sha256
sha3-512
sha384
sha512
```

These values are derived from the currently supported digests OpenSSL [OPENSSL]. Leading dashes were removed.

> The command `openssl dgst -list` (Version 1.1.1f from 2020-03-31) outputs the following:
>
> ```
> Supported digests:
> -blake2b512          -blake2s256          -md4
> -md5                 -md5-sha1            -ripemd
> -ripemd160           -rmd160              -sha1
> -sha224              -sha256              -sha3-224
> -sha3-256            -sha3-384            -sha3-512
> -sha384              -sha512              -sha512-224
> -sha512-256          -shake128            -shake256
> -sm3                 -ssl3-md5            -ssl3-sha1
> -whirlpool
> ```

The Value of the cryptographic hash representation (`value`) of value type `string` of 32 or more characters with `pattern` (regular expression):

```
^[0-9a-fA-F]{32,}$
```

The Value of the cryptographic hash attribute contains the cryptographic hash value in hexadecimal representation.

Examples 12:

```
37df33cb7464da5c7f077f4d56a32bc84987ec1d85b234537c1c1a4d4fc8d09dc29e2e762cb5203677bf849a2855a0283710f1f5fe1d6ce8d5ac85c645d0fcb
3
4775203615d9534a8bfca96a93dc8b461a489f69124a130d786b42204f3341cc
9ea4c8200113d49d26505da0e02e2f49055dc078d1ad7a419b32e291c7afebbb84badfbd46dec42883bea0b2a1fa697c
```

The filename representation (`filename`) of value type `string` with one or more characters contains the name of the file which is identified by the hash values.

Examples 13:

```
WINWORD.EXE
msotadddin.dll
sudoers.so
```

If the value of the hash matches and the filename does not, a user SHOULD prefer the hash value. In such cases, the filename SHOULD be used as informational property.

3.1.3.3.3 Full Product Name Type - Product Identification Helper - Model Numbers

The list of models (`model_numbers`) of value type `array` with 1 or more unique items contains a list of full or abbreviated (partial) model numbers.

A list of models SHOULD only be used if a certain range of model numbers with its corresponding software version is affected, or the model numbers change during update.

This can also be used to identify hardware. If necessary, the software, or any other related part, SHALL be bind to that via a product relationship.

```
"model_numbers": {
    //...
  "items": {
```

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

```
      //...
    }
  },
```

Any given model number of value type `string` with at least 1 character represents a full or abbreviated (partial) model number of the component to identify.

> The terms "model", "model number" and "model variant" are mostly used synonymously. Often it is abbreviated as "MN", M/N" or "model no.".

If a part of a model number of the component to identify is given, it SHOULD begin with the first character of the model number and stop at any point. Characters which SHOULD NOT be matched MUST be replaced by either `?` (for a single character) or `*` (for zero or more characters). Two `*` MUST NOT follow each other.

Examples 14:

```
6RA8096-4MV62-0AA0
6RA801?-??V62-0AA0
IC25T060ATCS05-0
```

3.1.3.3.4 Full Product Name Type - Product Identification Helper - PURL

The package URL (PURL) representation (`purl`) is a `string` of 7 or more characters with `pattern` (regular expression):

```
^pkg:[A-Za-z\\.\\-\\+][A-Za-z0-9\\.\\-\\+]*/.+
```

> The given pattern does not completely evaluate whether a PURL is valid according to the [PURL] specification. It provides a more generic approach and general guidance to enable forward compatibility. CSAF uses only the canonical form of PURL to conform with section 3.3 of [RFC3986]. Therefore, URLs starting with `pkg://` are considered invalid.

This package URL (PURL) attribute refers to a method for reliably identifying and locating software packages external to this specification. See [PURL] for details.

3.1.3.3.5 Full Product Name Type - Product Identification Helper - SBOM URLs

The list of SBOM URLs (`sbom_urls`) of value type `array` with 1 or more items contains a list of URLs where SBOMs for this product can be retrieved.

> The SBOMs might differ in format or depth of detail. Currently supported formats are SPDX, CycloneDX, and SWID.

```
"sbom_urls": {
    //...
  "items": {
    //...
  }
},
```

Any given SBOM URL of value type `string` with format `uri` contains a URL of one SBOM for this product.

Examples 15:

```
https://raw.githubusercontent.com/CycloneDX/bom-examples/master/SBOM/keycloak-10.0.2/bom.json
https://swinslow.net/spdx-examples/example4/main-bin-v2
```

3.1.3.3.6 Full Product Name Type - Product Identification Helper - Serial Numbers

The list of serial numbers (`serial_numbers`) of value type `array` with 1 or more unique items contains a list of full or abbreviated (partial) serial numbers.

A list of serial numbers SHOULD only be used if a certain range of serial numbers with its corresponding software version is affected, or the serial numbers change during update.

```
"serial_numbers": {
    //...
  "items": {
    //...
  }
```

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

```
    },
```

Any given serial number of value type `string` with at least 1 character represents a full or abbreviated (partial) serial number of the component to identify.

If a part of a serial number of the component to identify is given, it SHOULD begin with the first character of the serial number and stop at any point. Characters which SHOULD NOT be matched MUST be replaced by either `?` (for a single character) or `*` (for zero or more characters). Two `*` MUST NOT follow each other.

3.1.3.3.7 Full Product Name Type - Product Identification Helper - SKUs

The list of stock keeping units (`skus`) of value type `array` with 1 or more items contains a list of full or abbreviated (partial) stock keeping units.

A list of stock keeping units SHOULD only be used if the list of relationships is used to decouple e.g. hardware from the software, or the stock keeping units change during update. In the latter case the remediations SHALL include the new stock keeping units is or a description how it can be obtained.

> The use of the list of relationships in the first case is important. Otherwise, the end user is unable to identify which version (the affected or the not affected / fixed one) is used.

```
    "skus": {
        //...
      "items": {
        //...
      }
    },
```

Any given stock keeping unit of value type `string` with at least 1 character represents a full or abbreviated (partial) stock keeping unit (SKU) of the component to identify.

> Sometimes this is also called "item number", "article number" or "product number".

If a part of a stock keeping unit of the component to identify is given, it SHOULD begin with the first character of the stock keeping unit and stop at any point. Characters which SHOULD NOT be matched MUST be replaced by either `?` (for a single character) or `*` (for zero or more characters).
Two `*` MUST NOT follow each other.

3.1.3.3.8 Full Product Name Type - Product Identification Helper - Generic URIs

List of generic URIs (`x_generic_uris`) of value type `array` with at least 1 item contains a list of identifiers which are either vendor-specific or derived from a standard not yet supported.

```
    "x_generic_uris": {
      // ...
      "items": {
        // ...
      }
    }
```

Any such Generic URI item of value type `object` provides the two mandatory properties Namespace (`namespace`) and URI (`uri`).

```
        "properties": {
          "namespace": {
            // ...
          },
          "uri": {
            // ...
          }
        }
```

The namespace of the generic URI (`namespace`) of value type `string` with format `uri` refers to a URL which provides the name and knowledge about the specification used or is the namespace in which these values are valid.

The URI (`uri`) of value type `string` with format `uri` contains the identifier itself.

> These elements can be used to reference a specific component from an SBOM:

Example 16 linking a component from a CycloneDX SBOM using the bomlink mechanism:

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

```
          "x_generic_uris": [
            {
              "namespace": "https://cyclonedx.org/capabilities/bomlink/",
              "uri": "urn:cdx:411dafd2-c29f-491a-97d7-e97de5bc2289/1#pkg:maven/org.jboss.logging/jboss-
logging@3.4.1.Final?type=jar"
            }
          ]
```

Example 17 linking a component from an SPDX SBOM:

```
          "x_generic_uris": [
            {
              "namespace": "https://spdx.github.io/spdx-spec/document-creation-information/#65-spdx-document-namespace-
field",
              "uri": "https://swinslow.net/spdx-examples/example4/main-bin-v2#SPDXRef-libc"
            }
          ]
```

3.1.4 Language Type

Language type (`lang_t`) has value type `string` with `pattern` (regular expression):

```
    ^(([A-Za-z]{2,3}(-[A-Za-z]{3}(-[A-Za-z]{3}){0,2})?|[A-Za-z]{4,8})(-[A-Za-z]{4})?(-([A-Za-z]{2}|[0-9]{3}))?(-([A-Za-
z0-9]{5,8}|[0-9][A-Za-z0-9]{3}))*(-[A-WY-Za-wy-z0-9](-[A-Za-z0-9]{2,8})+)*(-[Xx](-[A-Za-z0-9]{1,8})+)?|[Xx](-[A-Za-z0-9]
{1,8})+|[Ii]-[Dd][Ee][Ff][Aa][Uu][Ll][Tt]|[Ii]-[Mm][Ii][Nn][Gg][Oo])$
```

The value identifies a language, corresponding to IETF BCP 47 / RFC 5646. See IETF language registry: https://www.iana.org/assignments/language-subtag-registry/language-subtag-registry

> CSAF skips those grandfathered language tags that are deprecated at the time of writing the specification. Even though the private use language tags are supported they should not be used to ensure readability across the ecosystem. It is recommended to follow the conventions for the capitalization of the subtags even though it is not mandatory as most users are used to that.

Examples 18:

```
    de
    en
    fr
    frc
    jp
```

3.1.5 Notes Type

List of notes (`notes_t`) of value type `array` with 1 or more items of type `Note` contains notes which are specific to the current context.

```
    "notes_t": {
      // ...
      "items": {
        // ...
      }
    },
```

Value type of every such Note item is `object` with the mandatory properties `category` and `text` providing a place to put all manner of text blobs related to the current context. A Note `object` MAY provide the optional properties `audience` and `title`.

```
    "properties": {
      "audience": {
        // ...
      },
      "category": {
        // ...
      },
      "text": {
        // ...
      },
```

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

```
    "title": {
      // ...
    }
  }
```

Audience of note (`audience`) of value type `string` with 1 or more characters indicates who is intended to read it.

Examples 19:

```
all
executives
operational management and system administrators
safety engineers
```

Note category (`category`) of value type `string` and `enum` contains the information of what kind of note this is. Valid `enum` values are:

```
description
details
faq
general
legal_disclaimer
other
summary
```

The value `description` indicates the note is a description of something. The optional sibling property `title` MAY have more information in this case.

The value `details` indicates the note is a low-level detailed discussion. The optional sibling property `title` MAY have more information in this case.

The value `faq` indicates the note is a list of frequently asked questions.

The value `general` indicates the note is a general, high-level note. The optional sibling property `title` MAY have more information in this case.

The value `legal_disclaimer` indicates the note represents any possible legal discussion, including constraints, surrounding the document.

The value `other` indicates the note is something that doesn't fit the other categories. The optional sibling attribute `title` SHOULD have more information to indicate clearly what kind of note to expect in this case.

The value `summary` indicates the note is a summary of something. The optional sibling property `title` MAY have more information in this case.

Note content (`text`) of value type `string` with 1 or more characters holds the content of the note. Content varies depending on type.

Title of note (`title`) of value type `string` with 1 or more characters provides a concise description of what is contained in the text of the note.

Examples 20:

```
Details
Executive summary
Technical summary
Impact on safety systems
```

3.1.6 Product Group ID Type

The Product Group ID Type (`product_group_id_t`) of value type `string` with 1 or more characters is a reference token for product group instances. The value is a token required to identify a group of products so that it can be referred to from other parts in the document. There is no predefined or required format for the Product Group ID (`product_group_id`) as long as it uniquely identifies a product group in the context of the current document.

```
"product_group_id_t": {
  // ...
},
```

Examples 21:

```
CSAFGID-0001
CSAFGID-0002
```

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

```
CSAFGID-0020
```

> Even though the standard does not require a specific format it is recommended to use different prefixes for the Product ID and the Product Group ID to support reading and parsing the document.

3.1.7 Product Groups Type

List of Product Group ID (`product_groups_t`) of value type `array` with 1 or more unique items (a `set`) of type Product Group ID (`product_group_id_t`) specifies a list of `product_group_ids` to give context to the parent item.

```
"product_groups_t": {
  // ...
  "items": {
    // ...
  }
},
```

3.1.8 Product ID Type

The Product ID Type (`product_id_t`) of value type `string` with 1 or more characters is a reference token for product instances. The value is a token required to identify a `full_product_name` so that it can be referred to from other parts in the document. There is no predefined or required format for the Product ID (`product_id`) as long as it uniquely identifies a product in the context of the current document.

```
"product_id_t": {
  // ...
},
```

Examples 22:

```
CSAFPID-0004
CSAFPID-0008
```

> Even though the standard does not require a specific format it is recommended to use different prefixes for the Product ID and the Product Group ID to support reading and parsing the document.

3.1.9 Products Type

List of Product IDs (`products_t`) of value type `array` with 1 or more unique items (a `set`) of type Product ID (`product_id_t`) specifies a list of `product_ids` to give context to the parent item.

```
"products_t": {
  // ...
  "items": {
    // ...
  }
},
```

3.1.10 References Type

List of references (`references_t`) of value type `array` with 1 or more items of type Reference holds a list of Reference objects.

```
"references_t": {
  // ...
  "items": {
    // ...
  }
},
```

Value type of every such Reference item is `object` with the mandatory properties `url` and `summary` holding any reference to conferences, papers, advisories, and other resources that are related and considered related to either a surrounding part of or the entire document and to be of value to the document consumer. A reference `object` MAY provide the optional property `category`.

```
"properties": {
  "category": {
    // ...
  },
```

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

```
    "summary": {
      // ...
    },
    "url": {
      // ...
    }
  }
```

Category of reference (`category`) of value type `string` and `enum` indicates whether the reference points to the same document or vulnerability in focus (depending on scope) or to an external resource. Valid `enum` values are:

```
external
self
```

The default value for `category` is `external`.

The value `external` indicates, that this document is an external reference to a document or vulnerability in focus (depending on scope).

The value `self` indicates, that this document is a reference to this same document or vulnerability (also depending on scope).

> This includes links to documents with the same content but different file format (e.g. advisories as PDF or HTML).

Summary of the reference (`summary`) of value type `string` with 1 or more characters indicates what this reference refers to.

URL of reference (`url`) of value type `string` with format `uri` provides the URL for the reference.

3.1.11 Version Type

The Version (`version_t`) type has value type `string` with `pattern` (regular expression):

```
^(0|[1-9][0-9]*)$|^((0|[1-9]\\d*)\\.(0|[1-9]\\d*)\\.(0|[1-9]\\d*)(?:-((?:0|[1-9]\\d*|\\d*[a-zA-Z-][0-9a-zA-Z-]*)
(?:\\.(?:0|[1-9]\\d*|\\d*[a-zA-Z-][0-9a-zA-Z-]*))*))?(?:\\+([0-9a-zA-Z-]+(?:\\.[0-9a-zA-Z-]+)*))?)$
```

The version specifies a version string to denote clearly the evolution of the content of the document. There are two options how it can be used:

- semantic versioning (preferred; according to the rules below)
- integer versioning

A CSAF document MUST use only one versioning system.

Examples 23:

```
1
4
0.9.0
1.4.3
2.40.0+21AF26D3
```

3.1.11.1 Version Type - Integer versioning

Integer versioning increments for each version where the `/document/tracking/status` is `final` the version number by one. The regular expression for this type is:

```
^(0|[1-9][0-9]*)$
```

The following rules apply:

1. Once a versioned document has been released, the contents of that version MUST NOT be modified. Any modifications MUST be released as a new version.
2. Version zero (0) is for initial development before the `initial_release_date`. The document status MUST be `draft`. Anything MAY change at any time. The document SHOULD NOT be considered stable.
3. Version 1 defines the initial public release. Each new version where `/document/tracking/status` is `final` has a version number incremented by one.
4. Pre-release versions (document status `draft`) MUST carry the new version number. Sole exception is before the initial release (see rule 2). The combination of document status `draft` and version 1 MAY be used to indicate that the content is unlikely to change.
5. Build metadata is never included in the version.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

6. Precedence MUST be calculate by integer comparison.

3.1.11.2 Version Type - Semantic versioning

Semantic versioning derived the rules from [SemVer]. The regular expression for this type is:

```
^((0|[1-9]\\d*)\\.(0|[1-9]\\d*)\\.(0|[1-9]\\d*)(?:-((?:0|[1-9]\\d*|\\d*[a-zA-Z-][0-9a-zA-Z-]*)(?:\\.(?:0|[1-9]\\d*|\\d*
[a-zA-Z-][0-9a-zA-Z-]*))*))?(?:\\+([0-9a-zA-Z-]+(?:\\.[0-9a-zA-Z-]+)*))?)$
```

The goal of this structure is to provide additional information to the end user whether a new comparison with the asset database is needed. The "public API" in regards to CSAF is the CSAF document with its structure and content. This results in the following rules:

1. A normal version number MUST take the form X.Y.Z where X, Y, and Z are non-negative integers, and MUST NOT contain leading zeroes. X is the major version, Y is the minor version, and Z is the patch version. Each element MUST increase numerically. For instance: 1.9.0 -> 1.10.0 -> 1.11.0.

2. Once a versioned document has been released, the contents of that version MUST NOT be modified. Any modifications MUST be released as a new version.

3. Major version zero (0.y.z) is for initial development before the `initial_release_date`. The document status MUST be `draft`. Anything MAY change at any time. The document SHOULD NOT be considered stable. Changes which would increment the major version according to rule 7 are tracked in this stage with (0.y.z) by incrementing the minor version y instead. Changes that would increment the minor or patch version according to rule 6 or 5 are both tracked in this stage with (0.y.z) by incrementing the patch version z instead.

4. Version 1.0.0 defines the initial public release. The way in which the version number is incremented after this release is dependent on the content and structure of the document and how it changes.

5. Patch version Z (x.y.Z | x > 0) MUST be incremented if only backwards compatible bug fixes are introduced. A bug fix is defined as an internal change that fixes incorrect behavior.

   > In the context of the document this is the case e.g. for spelling mistakes.

6. Minor version Y (x.Y.z | x > 0) MUST be incremented if the content of an existing element changes except for those which are covert through rule 7. It MUST be incremented if substantial new information are introduced or new elements are provided. It MAY include patch level changes. Patch version MUST be reset to 0 when minor version is incremented.

7. Major version X (X.y.z | X > 0) MUST be incremented if a new comparison with the end user's asset database is required. This includes:

   - changes (adding, removing elements or modifying content) in `/product_tree` or elements which contain `/product_tree` in their path
   - adding or removing items of `/vulnerabilities`
   - adding or removing elements in:
     - `/vulnerabilities[]/product_status/first_affected`
     - `/vulnerabilities[]/product_status/known_affected`
     - `/vulnerabilities[]/product_status/last_affected`
   - removing elements from:
     - `/vulnerabilities[]/product_status/first_fixed`
     - `/vulnerabilities[]/product_status/fixed`
     - `/vulnerabilities[]/product_status/known_not_affected`

   It MAY also include minor and patch level changes. Patch and minor version MUST be reset to 0 when major version is incremented.

8. A pre-release version (document status `draft`) MAY be denoted by appending a hyphen and a series of dot separated identifiers immediately following the patch version. Identifiers MUST comprise only ASCII alphanumerics and hyphens [0-9A-Za-z-]. Identifiers MUST NOT be empty. Numeric identifiers MUST NOT include leading zeroes. Pre-release versions have a lower precedence than the associated normal version. A pre-release version indicates that the version is unstable and might not satisfy the intended compatibility requirements as denoted by its associated normal version.

   Examples 24:

   ```
   1.0.0-0.3.7
   1.0.0-alpha
   1.0.0-alpha.1
   1.0.0-x-y-z.-
   1.0.0-x.7.z.92
   ```

9. Pre-release MUST NOT be included if `/document/tracking/status` is `final`.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

10. Build metadata MAY be denoted by appending a plus sign and a series of dot separated identifiers immediately following the patch or pre-release version. Identifiers MUST comprise only ASCII alphanumerics and hyphens [0-9A-Za-z-]. Identifiers MUST NOT be empty. Build metadata MUST be ignored when determining version precedence. Thus two versions that differ only in the build metadata, have the same precedence.

    Examples 25:

    ```
    1.0.0+20130313144700
    1.0.0+21AF26D3—-117B344092BD
    1.0.0-alpha+001
    1.0.0-beta+exp.sha.5114f85
    ```

11. Precedence refers to how versions are compared to each other when ordered.

    1. Precedence MUST be calculated by separating the version into major, minor, patch and pre-release identifiers in that order (Build metadata does not figure into precedence).
    2. Precedence is determined by the first difference when comparing each of these identifiers from left to right as follows: Major, minor, and patch versions are always compared numerically.

       Example 26:

       ```
       1.0.0 < 2.0.0 < 2.1.0 < 2.1.1
       ```

    3. When major, minor, and patch are equal, a pre-release version has lower precedence than a normal version:

       Example 27:

       ```
       1.0.0-alpha < 1.0.0
       ```

    4. Precedence for two pre-release versions with the same major, minor, and patch version MUST be determined by comparing each dot separated identifier from left to right until a difference is found as follows:

       1. Identifiers consisting of only digits are compared numerically.
       2. Identifiers with letters or hyphens are compared lexically in ASCII sort order.
       3. Numeric identifiers always have lower precedence than non-numeric identifiers.
       4. A larger set of pre-release fields has a higher precedence than a smaller set, if all of the preceding identifiers are equal.

       Example 28:

       ```
       1.0.0-alpha < 1.0.0-alpha.1 < 1.0.0-alpha.beta < 1.0.0-beta < 1.0.0-beta.2 < 1.0.0-beta.11 < 1.0.0-rc.1 <
       1.0.0
       ```

3.2 Properties

These final three subsections document the three properties of a CSAF document. The single mandatory property `document`, as well as the optional properties `product_tree` and `vulnerabilities` in that order.

3.2.1 Document Property

Document level meta-data (`document`) of value type `object` with the 5 mandatory properties Category (`category`), CSAF Version (`csaf_version`), Publisher (`publisher`), Title (`title`), and Tracking (`tracking`) captures the meta-data about this document describing a particular set of security advisories. In addition, the `document` object MAY provide the 7 optional properties Acknowledgments (`acknowledgments`), Aggregate Severity (`aggregate_severity`), Distribution (`distribution`), Language (`lang`), Notes (`notes`), References (`references`), and Source Language (`source_lang`).

```
"document": {
  // ...
  "properties": {
    "acknowledgments": {
      // ...
    },
    "aggregate_severity" : {
      // ...
    },
    "category": {
```

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

```
        // ...
      },
      "csaf_version": {
        // ...
      },
      "distribution": {
        // ...
      },
      "lang": {
        // ...
      },
      "notes": {
        // ...
      },
      "publisher": {
        // ...
      },
      "references": {
        // ...
      },
      "source_lang": {
        // ...
      },
      "title": {
        // ...
      },
      "tracking": {
        // ...
      }
    }
  },
```

3.2.1.1 Document Property - Acknowledgments

Document acknowledgments (`acknowledgments`) of value type Acknowledgments Type (`acknowledgments_t`) contains a list of acknowledgment elements associated with the whole document.

```
  "acknowledgments": {
    // ...
  },
```

3.2.1.2 Document Property - Aggregate Severity

Aggregate severity (`aggregate_severity`) of value type `object` with the mandatory property `text` and the optional property `namespace` is a vehicle that is provided by the document producer to convey the urgency and criticality with which the one or more vulnerabilities reported should be addressed. It is a document-level metric and applied to the document as a whole — not any specific vulnerability. The range of values in this field is defined according to the document producer's policies and procedures.

```
  "aggregate_severity": {
    // ...
    "properties": {
      "namespace": {
        // ...
      },
      "text": {
        // ...
      }
    }
  },
```

The Namespace of aggregate severity (`namespace`) of value type `string` with format `uri` points to the namespace so referenced.

The Text of aggregate severity (`text`) of value type `string` with 1 or more characters provides a severity which is independent of - and in addition to - any other standard metric for determining the impact or severity of a given vulnerability (such as CVSS).

Examples 29:

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

```
Critical
Important
Moderate
```

3.2.1.3 Document Property - Category

Document category (`category`) with value type `string` of 1 or more characters with `pattern` (regular expression):

```
^[^\\s\\-_\\.](.*[^\\s\\-_\\.])?$
```

Document category defines a short canonical name, chosen by the document producer, which will inform the end user as to the category of document.

> It is directly related to the profiles defined in section 4.

```
"category": {
  // ...
}
```

Examples 30:

```
csaf_base
csaf_security_advisory
csaf_vex
Example Company Security Notice
```

3.2.1.4 Document Property - CSAF Version

CSAF version (`csaf_version`) of value type `string` and `enum` gives the version of the CSAF specification which the document was generated for. The single valid value for this `enum` is:

```
2.0
```

3.2.1.5 Document Property - Distribution

Rules for sharing document (`distribution`) of value type `object` with at least 1 of the 2 properties Text (`text`) and Traffic Light Protocol (TLP) (`tlp`) describes any constraints on how this document might be shared.

```
"distribution": {
  // ...
  "properties": {
    "text": {
      // ...
    },
    "tlp": {
      // ...
    }
  }
},
```

If both values are present, the TLP information SHOULD be preferred as this aids in automation.

3.2.1.5.1 Document Property - Distribution - Text

The Textual description (`text`) of value type `string` with 1 or more characters provides a textual description of additional constraints.

Examples 31:

```
Copyright 2021, Example Company, All Rights Reserved.
Distribute freely.
Share only on a need-to-know-basis only.
```

3.2.1.5.2 Document Property - Distribution - TLP

Traffic Light Protocol (TLP) (`tlp`) of value type `object` with the mandatory property Label (`label`) and the optional property URL (`url`) provides details about the TLP classification of the document.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025

```
"tlp": {
  // ...
  "properties": {
    "label": {
      // ...
    },
    "url": {
      // ...
    }
  }
}
```

The Label of TLP (`label`) with value type `string` and `enum` provides the TLP label of the document. Valid values of the `enum` are:

```
AMBER
GREEN
RED
WHITE
```

The URL of TLP version (`url`) with value type `string` with format `uri` provides a URL where to find the textual description of the TLP version which is used in this document. The default value is the URL to the definition by FIRST:

```
https://www.first.org/tlp/
```

Examples 32:

```
https://www.us-cert.gov/tlp
https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Kritis/Merkblatt_TLP.pdf
```

3.2.1.6 Document Property - Language

Document language (`lang`) of value type Language Type (`lang_t`) identifies the language used by this document, corresponding to IETF BCP 47 / RFC 5646.

3.2.1.7 Document Property - Notes

Document notes (`notes`) of value type Notes Type (`notes_t`) holds notes associated with the whole document.

```
"notes": {
  // ...
},
```

3.2.1.8 Document Property - Publisher

Publisher (`publisher`) has value type `object` with the mandatory properties Category (`category`), Name (`name`) and Namespace (`namespace`) and provides information on the publishing entity. The 2 other optional properties are: `contact_details` and `issuing_authority`.

```
"publisher": {
  // ...
  "properties": {
    "category": {
      // ...
    },
    "contact_details": {
      // ...
    },
    "issuing_authority": {
      // ...
    },
    "name": {
      // ...
    }
    "namespace": {
      // ...
    }
  }
```

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20153:2025