STIX[TM] Version 1.2.1. Part 13: Data Marking

Committee Specification Draft 01 / Public Review Draft 01

06 November 2015

Specification URIs

This version:

http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part13-data-marking/stix-v1.2.1-csprd01-part13-data-marking.docx (Authoritative)
http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part13-data-marking/stix-v1.2.1-csprd01-part13-data-marking.html
http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part13-data-marking/stix-v1.2.1-csprd01-part13-data-marking.pdf

Previous version:

N/A

Latest version:

http://docs.oasis-open.org/cti/stix/v1.2.1/stix-v1.2.1-part13-data-marking.docx (Authoritative)
http://docs.oasis-open.org/cti/stix/v1.2.1/stix-v1.2.1-part13-data-marking.html
http://docs.oasis-open.org/cti/stix/v1.2.1/stix-v1.2.1-part13-data-marking.pdf

Technical Committee:

OASIS Cyber Threat Intelligence (CTI) TC

Chair:

Richard Struse (Richard.Struse@HQ.DHS.GOV), DHS Office of Cybersecurity and Communications (CS&C)

Editors:

Sean Barnum (sbarnum@mitre.org), MITRE Corporation
Desiree Beck (dbeck@mitre.org), MITRE Corporation
Aharon Chernin (achernin@soltra.com), Soltra
Rich Piazza (rpiazza@mitre.org), MITRE Corporation

Additional artifacts:

This prose specification is one component of a Work Product that also includes:

- *STIX Version 1.2.1. Part 1: Overview*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part1-overview/stix-v1.2.1-csprd01-part1-overview.html
- *STIX Version 1.2.1. Part 2: Common*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part2-common/stix-v1.2.1-csprd01-part2-common.html
- *STIX Version 1.2.1. Part 3: Core*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part3-core/stix-v1.2.1-csprd01-part3-core.html
- *STIX Version 1.2.1. Part 4: Indicator*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part4-indicator/stix-v1.2.1-csprd01-part4-indicator.html
- *STIX Version 1.2.1. Part 5: TTP*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part5-ttp/stix-v1.2.1-csprd01-part5-ttp.html
- *STIX Version 1.2.1. Part 6: Incident*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part6-incident/stix-v1.2.1-csprd01-part6-incident.html

- *STIX Version 1.2.1. Part 7: Threat Actor*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part7-threat-actor/stix-v1.2.1-csprd01-part7-threat-actor.html
- *STIX Version 1.2.1. Part 8: Campaign*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part8-campaign/stix-v1.2.1-csprd01-part8-campaign.html
- *STIX Version 1.2.1. Part 9: Course of Action*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part9-coa/stix-v1.2.1-csprd01-part9-coa.html
- *STIX Version 1.2.1. Part 10: Exploit Target*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part10-exploit-target/stix-v1.2.1-csprd01-part10-exploit-target.html
- *STIX Version 1.2.1. Part 11: Report*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part11-report/stix-v1.2.1-csprd01-part11-report.html
- *STIX Version 1.2.1. Part 12: Default Extensions*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part12-extensions/stix-v1.2.1-csprd01-part12-extensions.html
- *STIX Version 1.2.1. Part 13: Data Marking* (this document). http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part13-data-marking/stix-v1.2.1-csprd01-part13-data-marking.html
- *STIX Version 1.2.1. Part 14: Vocabularies*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part14-vocabularies/stix-v1.2.1-csprd01-part14-vocabularies.html
- *STIX Version 1.2.1. Part 15: UML Model*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part15-uml-model/stix-v1.2.1-csprd01-part15-uml-model.html
- UML Model Serialization: http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/uml-model/

Related work:

This specification replaces or supersedes:

- *STIX™ 1.2 Data Marking Specification (v1.2)* https://github.com/STIXProject/specifications/blob/version1.2/documents/pdf%20versions/STIX_DataMarking_Draft.pdf

This specification is related to:

- *CybOX™ Version 2.1.1.* Work in progress. https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=cti-cybox
- *CybOX™ 2.1.* https://cyboxproject.github.io/

Abstract:

The Structured Threat Information Expression (STIX) framework defines nine core constructs and the relationships between them for the purposes of modeling cyber threat information and enabling cyber threat information analysis and sharing. This specification document defines the Data Marking data model, which provides an independent, flexible, structured capability for data marking expression.

Status:

This document was last revised or approved by the OASIS Cyber Threat Intelligence (CTI) TC on the above date. The level of approval is also listed above. Check the "Latest version" location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Technical Committee (TC) are listed at https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=cti#technical.

TC members should send comments on this specification to the TC's email list. Others should send comments to the TC's public comment list, after subscribing to it by following the instructions at the "Send A Comment" button on the TC's web page at https://www.oasis-open.org/committees/cti/.

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/cti/ipr.php).

Citation format:

When referencing this specification the following citation format should be used:

[STIX-v1.2.1-Data-Marking]

STIX[TM] Version 1.2.1. Part 13: Data Marking. Edited by Sean Barnum, Desiree Beck, Aharon Chernin, and Rich Piazza. 06 November 2015. OASIS Committee Specification Draft 01 / Public Review Draft 01. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part13-data-marking/stix-v1.2.1-csprd01-part13-data-marking.html. Latest version: http://docs.oasis-open.org/cti/stix/v1.2.1/stix-v1.2.1-part13-data-marking.html.

Notices

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

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

This document and the information contained herein is provided on an "AS IS" basis and OASIS 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.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see https://www.oasis-open.org/policies-guidelines/trademark for above guidance.

STIX[™], TAXII[™], AND CybOX[™] (STANDARD OR STANDARDS) AND THEIR COMPONENT PARTS ARE PROVIDED “AS IS” WITHOUT ANY WARRANTY OF ANY KIND, EITHER EXPRESSED, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTY THAT THESE STANDARDS OR ANY OF THEIR COMPONENT PARTS WILL CONFORM TO SPECIFICATIONS, ANY

IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR FREEDOM FROM INFRINGEMENT, ANY WARRANTY THAT THE STANDARDS OR THEIR COMPONENT PARTS WILL BE ERROR FREE, OR ANY WARRANTY THAT THE DOCUMENTATION, IF PROVIDED, WILL CONFORM TO THE STANDARDS OR THEIR COMPONENT PARTS. IN NO EVENT SHALL THE UNITED STATES GOVERNMENT OR ITS CONTRACTORS OR SUBCONTRACTORS BE LIABLE FOR ANY DAMAGES, INCLUDING, BUT NOT LIMITED TO, DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES, ARISING OUT OF, RESULTING FROM, OR IN ANY WAY CONNECTED WITH THESE STANDARDS OR THEIR COMPONENT PARTS OR ANY PROVIDED DOCUMENTATION, WHETHER OR NOT BASED UPON WARRANTY, CONTRACT, TORT, OR OTHERWISE, WHETHER OR NOT INJURY WAS SUSTAINED BY PERSONS OR PROPERTY OR OTHERWISE, AND WHETHER OR NOT LOSS WAS SUSTAINED FROM, OR AROSE OUT OF THE RESULTS OF, OR USE OF, THE STANDARDS, THEIR COMPONENT PARTS, AND ANY PROVIDED DOCUMENTATION. THE UNITED STATES GOVERNMENT DISCLAIMS ALL WARRANTIES AND LIABILITIES REGARDING THE STANDARDS OR THEIR COMPONENT PARTS ATTRIBUTABLE TO ANY THIRD PARTY, IF PRESENT IN THE STANDARDS OR THEIR COMPONENT PARTS AND DISTRIBUTES IT OR THEM “AS IS.”

Table of Contents

1 Introduction

[All text is normative unless otherwise labeled]

The Structured Threat Information Expression (STIX[TM]) framework defines nine top-level component data models: Observable[1], Indicator, Incident, TTP, ExploitTarget, CourseOfAction, Campaign, ThreatActor, and Report. In addition, it defines a data model that captures data marking information for STIX content. This document serves as the specification for the STIX Data Marking data model.

Given the potentially sensitive nature of cyber threat information, a consistent requirement across many of the STIX component data models is the ability to represent markings of the data to specify things such as handling restrictions, terms of use, or copyright information. There currently exists no broad consensus standardized approach for such data markings; instead, there are various approaches within differing communities, driven by different motivations and usage contexts. Therefore, rather than adopting a single marking approach and expecting all STIX users to accept it, STIX takes a flexible and generic approach through the definition of the Data Marking data model.

In Section **1.1** we discuss additional specification documents, in Section **1.2** we provide document conventions, and in Section **1.3** we provide terminology. References are given in Sections **1.4** and **1.5**. In Section **2**, we give background information to help the reader better understand the specification details that are provided later in the document. We present the Data Marking data model specification details in Section **3** and conformance information in Section **4**.

1.1 STIX[TM] Specification Documents

The STIX specification consists of a formal UML model and a set of textual specification documents that explain the UML model. Specification documents have been written for each of the key individual data models that compose the full STIX UML model.

The *STIX Version 1.2.1 Part 1: Overview* document provides a comprehensive overview of the full set of STIX data models, which in addition to the nine top-level component data models mentioned in the Introduction, includes a core data model, a common data model, a cross-cutting data marking data model, various extension data models, and a set of default controlled vocabularies. *STIX Version 1.2.1 Part 1: Overview* also summarizes the relationship of STIX to other languages, and outlines general STIX data model conventions.

Figure 1-1 illustrates the set of specification documents that are available. The color black is used to indicate the specification overview document, altered shading differentiates the overarching Core and Common data models from the supporting data models (vocabularies, data marking, and default extensions), and the color white indicates the component data models. The solid grey color denotes the overall STIX Language UML model. This Data Marking specification document is highlighted in its associated color (see Section **1.2.3.3**). For a list of all STIX documents and related information sources, please see *STIX Version 1.2.1 Part 1: Overview*.

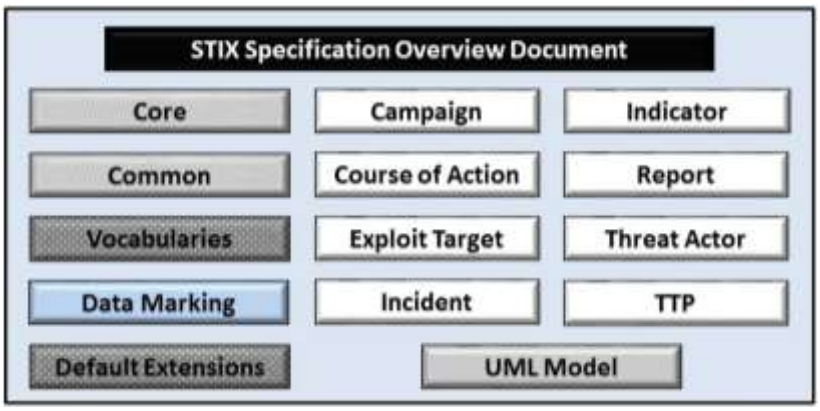


Figure 1-1. STIX[TM] Language v1.2.1 specification documents

1.2 Document Conventions

The following conventions are used in this document.

1.2.1 Fonts

The following font and font style conventions are used in the document:

- Capitalization is used for STIX high level concepts, which are defined in *STIX Version 1.2.1 Part 1: Overview*.

  Examples: Indicator, Course of Action, Threat Actor

- The `Courier New` font is used for writing UML objects.

  Examples: `RelatedIndicatorsType`, `stixCommon:StatementType`

  Note that all high level concepts have a corresponding UML object. For example, the Course of Action high level concept is associated with a UML class named, `CourseOfActionType`.

- The '*italic*' font (with single quotes) is used for noting actual, explicit values for STIX Language properties. The *italic* font (without quotes) is used for noting example values.

  Example: *'PackageIntentVocab-1.0,' high, medium, low*

1.2.2 UML Package References

Each STIX data model is captured in a different UML package (e.g., Core package, Campaign package, etc.) where the packages together compose the full STIX UML model. To refer to a particular class of a specific package, we use the format `package_prefix:class`, where `package_prefix` corresponds to the appropriate UML package. *STIX Version 1.2.1 Part 1: Overview* contains a list of the packages used by the Data Marking data model, along with the associated prefix notations, descriptions, examples.

Note that in this specification document, we do not explicitly specify the package prefix for any classes that originate from the Data Marking data model.

1.2.3 UML Diagrams

This specification makes use of UML diagrams to visually depict relationships between STIX Language constructs. Note that the diagrams have been extracted directly from the full UML model for STIX; they

have not been constructed purely for inclusion in the specification documents. Typically, diagrams are included for the primary class of a data model, and for any other class where the visualization of its relationships between other classes would be useful. This implies that there will be very few diagrams for classes whose only properties are either a data type or a class from the STIX Common data model. Other diagrams that are included correspond to classes that specialize a superclass and abstract or generalized classes that are extended by one or more subclasses.

In UML diagrams, classes are often presented with their attributes elided, to avoid clutter. A class presented with an empty section at the bottom of the icon indicates that there are no attributes other than those that are visualized using associations.

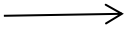
1.2.3.1 Class Properties

Generally, a class property can be shown in a UML diagram as either an attribute or an association (i.e., the distinction between attributes and associations is somewhat subjective). In order to make the size of UML diagrams in the specifications manageable, we have chosen to capture most properties as attributes and to capture only higher level properties as associations, especially in the main top-level component diagrams. In particular, we will always capture properties of UML data types as attributes. For example, properties of a class that are identifiers, titles, and timestamps will be represented as attributes.

1.2.3.2 Diagram Icons and Arrow Types

Diagram icons are used in a UML diagram to indicate whether a shape is a class, enumeration, or a data type, and decorative icons are used to indicate whether an element is an attribute of a class or an enumeration literal. In addition, two different arrow styles indicate either a directed association relationship (regular arrowhead) or a generalization relationship (triangle-shaped arrowhead). The icons and arrow styles we use are shown and described in **Table 1-1**.

Table 1-1. UML diagram icons

Icon	Description
	This diagram icon indicates a class. If the name is in italics, it is an abstract class.
	This diagram icon indicates an enumeration.
	This diagram icon indicates a data type.
	This decorator icon indicates an attribute of a class. The green circle means its visibility is public. If the circle is red or yellow, it means its visibility is private or protected.
	This decorator icon indicates an enumeration literal.
	This arrow type indicates a directed association relationship.
	This arrow type indicates a generalization relationship.

1.2.3.3 Color Coding

The shapes of the UML diagrams are color coded to indicate the data model associated with a class. The colors used in the Data Marking specification are illustrated via exemplars in **Figure 1-2**.

Figure 1-2. Data model color coding

1.2.4 Property Table Notation

Throughout Section **3**, tables are used to describe the properties of each data model class. Each property table consists of a column of names to identify the property, a type column to reflect the datatype of the property, a multiplicity column to reflect the allowed number of occurrences of the property, and a description column that describes the property. Package prefixes are provided for classes outside of the Data Marking data model (see Section **1.2.2**).

Note that if a class is a specialization of a superclass, only the properties that constitute the specialization are shown in the property table (i.e., properties of the superclass will not be shown). However, details of the superclass may be shown in the UML diagram.

In addition, properties that are part of a "choice" relationship (e.g., Prop1 OR Prop2 is used but not both) will be denoted by a unique letter subscript (e.g., API_Call$_A$, Code$_B$) and single logic expression in the Multiplicity column. For example, if there is a choice of property `API_Call`$_A$ and `Code`$_B$, the expression "A(1)|B(0..1)" will indicate that the `API_Call` property can be chosen with multiplicity 1 or the `Code` property can be chosen with multiplicity 0 or 1.

1.2.5 Property and Class Descriptions

Each class and property defined in STIX is described using the format, "The X property verb Y." For example, in the specification for the STIX Campaign, we write, "The `id` property specifies a globally unique identifier for the Campaign instance." In fact, the verb "specifies" could have been replaced by any number of alternatives: "defines," "describes," "contains," "references," etc.

However, we thought that using a wide variety of verb phrases might confuse a reader of a specification document because the meaning of each verb could be interpreted slightly differently. On the other hand, we didn't want to use a single, generic verb, such as "describes," because although the different verb choices may or may not be meaningful from an implementation standpoint, a distinction could be useful to those interested in the modeling aspect of STIX.

Consequently, we have chosen to use the three verbs, defined as follows, in class and property descriptions:

Verb	STIX Definition
captures	Used to record and preserve information without implying anything about the structure of a class or property. Often used for properties that encompass general content. This is the least precise of the three verbs.
	Examples: The `Source` property characterizes the source of the sighting information. Examples of details captured include identitifying characteristics, time-related attributes, and a list of the tools used to collect the information. The `Description` property captures a textual description of the Indicator.

characterizes	Describes the distinctive nature or features of a class or property. Often used to describe classes and properties that themselves comprise one or more other properties.
	Example: The `Confidence` property characterizes the level of confidence in the accuracy of the overall content captured in the Incident. The `ActivityType` class characterizes basic information about an activity a defender might use in response to a Campaign.
specifies	Used to clearly and precisely identify particular instances or values associated with a property. Often used for properties that are defined by a controlled vocabulary or enumeration; typically used for properties that take on only a single value.
	Example: The `version` property specifies the version identifier of the STIX Campaign data model used to capture the information associated with the Campaign.

1.3 Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in **[RFC2119]**.

1.4 Normative References

[RFC2119] Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels”, BCP 14, RFC 2119, March 1997. http://www.ietf.org/rfc/rfc2119.txt.

1.5 Non-Normative References

[TLP] Traffic Light Protocol (TLP) Matrix and Frequently Asked Questions. (n.d.). US-CERT. [Online]. http://www.us-cert.gov/tlp. Accessed Sep. 2, 2015.

2 Background

In this section, we provide high level information about the Data Marking data model that is necessary to fully understand the Data Marking data model specification details given in Section **3**. As explained in the introduction, the data marking construct is not conveyed as a separate entity in the STIX architecture diagram like the nine component data models; instead, the data marking construct exists as a cross-cutting structure across all of those constructs.

2.1 Marking Approach

There are two aspects to the STIX approach to data marking: (1) a controlled structure is used to specify the set of STIX content elements to which data markings apply, and (2) a marking structure is used to specify the particular data markings that are applied to the set of elements identified by the controlled structure.

This approach makes STIX data marking flexible in two ways. First, it permits the use of *any* data marking structure simply as a specialization of the Data Marking base class (the `MarkingStructureType` class; see Section **3.3**). Second, data marking information is specified separately from the STIX content being marked: instead of embedding the marking information within an individual property, property locations are *referenced* from a higher level (the `Controlled_Structure` property of the `MarkingType` class; see Section **3.2**). This makes data marking information space efficient and easier to update and refine, and it also enables any given STIX content to be marked with multiple marking schemes. Any level of information can be marked: individual properties, an entire STIX document, or anything in between[2]. For example, a copyright may be applied across a whole document while specific terms of use might apply only to certain properties of Indicator test mechanisms.

2.2 Using Markings

Before discussing about how markings are represented in STIX, it may be useful to understand how and where markings are used. The most common place to see data markings is in the `Handling` property of the STIX Header (`STIXHeaderType` class). Markings placed in this property are often used to apply markings globally, either to the entire STIX Package or to specific types of information regardless of where they appear in the STIX Package. For example, a copyright that applies to the entire STIX Package would be best placed in the `Handling` property of the STIX Header. Similarly, the indication that *all* Indicator Courses of Action are TLP:RED (see **[TLP]**) would also be best placed in the STIX Header.

However, the STIX Header is not the only place where data markings can be used. Individual STIX components (Indicators, Campaigns, etc.) all have their own `Handling` property, which if used restricts the marking applicability to just the properties within that component. This allows consumers to safely preserve markings within a component and move it between documents or into a datastore without worrying that the markings will change in meaning. Note that if the Handling property is placed directly in an individual component (e.g., `IndicatorType` class) rather than in a STIX Header, the `Handling` property name is still of type `marking:MarkingType` because the Data Marking data model provides a common structure, regardless of where data markings are used.

3 STIX[TM] Data Marking Data Model

The STIX Data Marking data model defines three classes used to capture data marking information for STIX content. Each of these classes is defined below.

3.1 MarkingType Class

The `MarkingType` class specifies a set of zero or more data marking specifications to be applied to the STIX content.

The property table for the `MarkingType` class is shown in **Table 3-1**.

Table 3-1. Properties of the `MarkingType` class

Name	Type	Multiplicity	Description
Marking	`MarkingSpecificationType`	0..*	The `Marking` property characterizes a data marking specification that is applied to STIX content. Information captured includes the structured elements to which the data marking is to be applied, a set of marking structures, and source information.

3.2 MarkingSpecificationType Class

The `MarkingSpecificationType` class characterizes a data marking specification that is applied to the STIX content. Information captured includes the structured elements to which the data marking is to be applied, a set of marking structures, and source information.

The UML diagram of the `MarkingSpecificationType` class is shown in **Figure 3-1**.

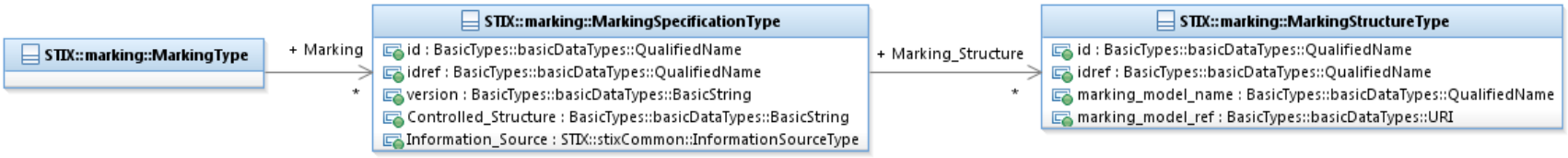


Figure 3-1. UML diagram of the `MarkingSpecificationType` class

The property table of the MarkingSpecificationType class that corresponds to **Figure 3-1** is given in **Table 3-2**.

Table 3-2. Properties of the MarkingSpecificationType class

Name	Type	Multiplicity	Description
id	basicDataTypes:QualifiedName	0..1	The id property specifies a globally unique identifier for a data marking specification instance.
idref	basicDataTypes:QualifiedName	0..1	The idref property specifies a reference to the identifier of a data marking specification instance specified elsewhere; the referenced data marking should be evaluated as if it were located where the data marking reference is defined. When the idref property is used, the id property MUST NOT also be specified and the other properties of the MarkingSpecificationType class SHOULD NOT hold any content.
version	basicDataTypes:BasicString	0..1	The version property specifies the version number of the STIX Data Marking data model used to capture the data marking associated with the STIX content.
Controlled_Structure	basicDataTypes:BasicString	0..1	The Controlled_Structure property specifies the full explicit set of STIX structured elements to which the marking is to be applied. The controlled structure MUST explicitly select *all* structured elements that the marking applies to; selecting a parent structured element may not imply that the marking also applies to its children. Specific syntax for how the set of STIX structured elements will be specified is dependent on the particular syntactic implementation (XML, JSON, etc.) of the STIX language and MUST be explicitly specified in a separate binding specification for that syntactic implementation (e.g. a STIX XML Binding Specification). For example, a STIX XML Binding Specification could specify XPath 1.0[3] as an appropriate choice for the syntax of the Controlled_Structure property.
Marking_Structure	MarkingStructureType	0..*	The Marking_Structure property characterizes the marking information to be applied to a portion of STIX content as specified in the Controlled_Structure property. Its underlying class is intended to be extended to enable the

			expression of any structured or unstructured data marking mechanism.
Information_Source	`stixCommon:InformationSourceType`	0..1	The `Information_Source` property characterizes the source of the data marking specification information. Examples of details captured include identifying characteristics (e.g., who marked the data) and time-related attributes (e.g., when the data was marked).

3.3 MarkingStructureType Class

The `MarkingStructureType` class characterizes the marking information to be applied to STIX content. The class is simply a mechanism for leveraging externally defined marking systems, and it is intended to be extended to enable the expression of any structured or unstructured data marking mechanism.

As illustrated in **Figure 3-2**, STIX v1.2.1 defines default subclasses for three particular data marking formats: Simple, Traffic Light Protocol (TLP), and Terms of Use (qualified names are not shown in the figure due to space considerations). See *STIX Version 1.2.1 Part 12: Default Extensions* for details. Producers who want to use another marking system may simply define a new extension to the `MarkingStructureType` class.

It is valid to mark a structured element with multiple markings from the same system or mark a structured element across multiple marking systems. If a structured element is marked multiple times using the same marking system, that system (not STIX) is responsible for specifying the semantic meaning of multiple markings, and if necessary, for specifying how conflicts should be resolved. If a structured element is marked across multiple marking systems, each system is considered individually applicable. If there are conflicting markings across marking systems the behavior is undefined; therefore, producers should make every effort to ensure documents are marked consistently and correctly among all marking systems. The data marking systems themselves should also define the interpretation of unmarked structured elements.

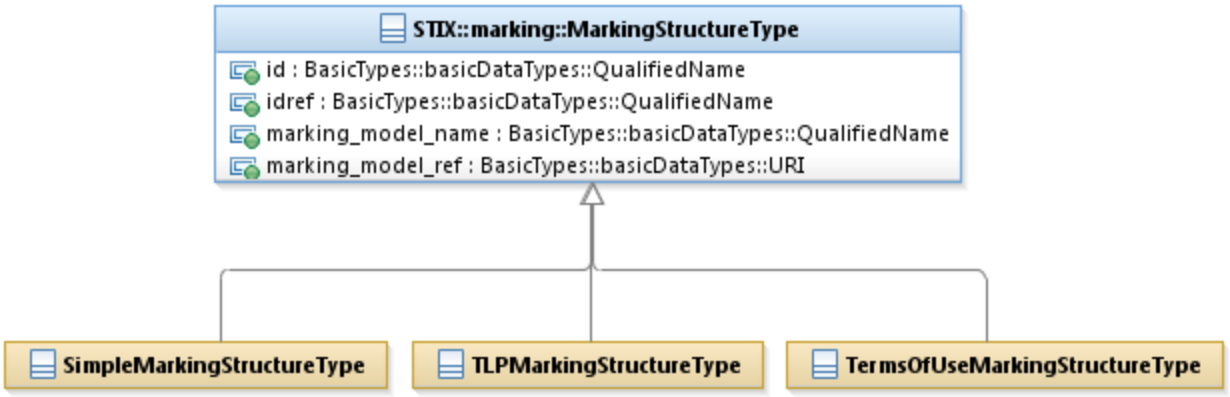


Figure 3-2. UML diagram of the `MarkingStructureType` class

As listed in **Table 3-3**, the three default subclasses and their descriptions are defined as possible extensions to the `MarkingStructureType` class. As stated above, additional markings can be used by defining a new subclass of the `MarkingStructureType` class. *STIX Version 1.2.1 Part 12: Default Extensions* gives further details of each extension shown.

Table 3-3. Default extensions of the `MarkingStructureType` class

Subclasses	Description
SimpleMarkingStructureType	The Simple marking structure allows users to make a text statement to mark the content. For example, copyright information can be communicated.
TLPMarkingStructureType	The Traffic Light Protocol (TLP) marking structure indicates how content may be shared. TLP statements are indicated through the use of a simple enumeration.
TermsOfUseMarkingStructureType	The Terms of Use marking structure allows users to make a text statement to specify the terms of use of the marked content. This marking is similar to the Simple marking structure, but it has stronger semantic meaning.

To reiterate, the `MarkingStructureType` class is simply a mechanism for leveraging externally defined marking systems. The data marking systems themselves define the semantics of what the markings mean, how multiple markings to the same structured element should be applied, and what to do if a structured element is unmarked. The `MarkingStructureType` class can be used to mark the data with anything. For example, data markings could be used to indicate that the STIX document is part of an exercise and is not actual production data.

The properties of the `MarkingStructureType` class are given in **Table 3-4**.

Table 3-4. Properties of the `MarkingStructureType` class

Name	Type	Multiplicity	Description
id	`basicDataType:QualifiedName`	0..1	The `id` property specifies a globally unique identifier for the marking structure instance.
idref	`basicDataType:QualifiedName`	0..1	The `idref` property specifies a reference to the identifier of a marking structure instance specified elsewhere; the referenced data marking should be evaluated as if it were located where the data marking reference is defined. When `idref` is specified, the `id` property MUST NOT also be specified, any other properties of the `MarkingStructureType` class SHOULD NOT hold any content, and the `MarkingStructureType` class SHOULD NOT be extended.
marking_model_name	`basicDataType:QualifiedName`	0..1	The `marking_model_name` property specifies the name of the marking model to be applied within the marking structure.
marking_model_ref	`basicDataType:URI`	0..1	The `marking_model_ref` property specifies a reference URI for the location of the authoriative descriptive source on the marking model to be applied within the marking structure.

4 Conformance

Implementations have discretion over which parts (components, properties, extensions, controlled vocabularies, etc.) of STIX they implement (e.g., Indicator/Suggested_COAs).

[1] Conformant implementations must conform to all normative structural specifications of the UML model or additional normative statements within this document that apply to the portions of STIX they implement (e.g., Implementers of the entire TTP component must conform to all normative structural specifications of the UML model or additional normative statements within this document regarding the TTP component).

[2] Conformant implementations are free to ignore normative structural specifications of the UML model or additional normative statements within this document that do not apply to the portions of STIX they implement (e.g., Non-implementers of any particular properties of the TTP component are free to ignore all normative structural specifications of the UML model or additional normative statements within this document regarding those properties of the TTP component).

The conformance section of this document is intentionally broad and attempts to reiterate what already exists in this document. The STIX 1.2 Specifications, which this specification is based on, did not have a conformance section. Instead, the STIX 1.2 Specifications relied on normative statements and the non-mandatory implementation of STIX profiles. STIX 1.2.1 represents a minimal change from STIX 1.2, and in that spirit no requirements have been added, modified, or removed by this section.

Appendix A. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

Dean Thompson, Australia and New Zealand Banking Group (ANZ Bank)
Bret Jordan, Blue Coat Systems, Inc.
Adnan Baykal, Center for Internet Security (CIS)
Jyoti Verma, Cisco Systems
Liron Schiff, Comilion (mobile) Ltd.
Jane Ginn, Cyber Threat Intelligence Network, Inc. (CTIN)
Richard Struse, DHS Office of Cybersecurity and Communications (CS&C)
Marlon Taylor, DHS Office of Cybersecurity and Communications (CS&C)
David Eilken, Financial Services Information Sharing and Analysis Center (FS-ISAC)
Sarah Brown, Fox-IT
Ryusuke Masuoka, Fujitsu Limited
Eric Burger, Georgetown University
Jason Keirstead, IBM
Paul Martini, iboss, Inc.
Jerome Athias, Individual
Terry MacDonald, Individual
Alex Pinto, Individual
Patrick Maroney, Integrated Networking Technologies, Inc.
Wouter Bolsterlee, Intelworks BV
Joep Gommers, Intelworks BV
Sergey Polzunov, Intelworks BV
Rutger Prins, Intelworks BV
Andrei Sîrghi, Intelworks BV
Raymon van der Velde, Intelworks BV
Jonathan Baker, MITRE Corporation
Sean Barnum, MITRE Corporation
Desiree Beck, MITRE Corporation
Mark Davidson, MITRE Corporation
Ivan Kirillov, MITRE Corporation
Jon Salwen, MITRE Corporation
John Wunder, MITRE Corporation
Mike Boyle, National Security Agency
Jessica Fitzgerald-McKay, National Security Agency
Takahiro Kakumaru, NEC Corporation
John-Mark Gurney, New Context Services, Inc.
Christian Hunt, New Context Services, Inc.
Daniel Riedel, New Context Services, Inc.
Andrew Storms, New Context Services, Inc.
John Tolbert, Queralt, Inc.
Igor Baikalov, Securonix
Bernd Grobauer, Siemens AG
Jonathan Bush, Soltra
Aharon Chernin, Soltra
Trey Darley, Soltra
Paul Dion, Soltra
Ali Khan, Soltra
Natalie Suarez, Soltra
Cedric LeRoux, Splunk Inc.
Brian Luger, Splunk Inc.

Crystal Hayes, The Boeing Company
Brad Butts, U.S. Bank
Mona Magathan, U.S. Bank
Adam Cooper, United Kingdom Cabinet Office
Mike McLellan, United Kingdom Cabinet Office
Chris O'Brien, United Kingdom Cabinet Office
Julian White, United Kingdom Cabinet Office
Anthony Rutkowski, Yaana Technologies, LLC

The authors would also like to thank the larger STIX Community for its input and help in reviewing this document.

Appendix B. Revision History

Revision	Date	Editor	Changes Made
wd01	21 August 2015	Sean Barnum Desiree Beck Aharon Chernin Rich Piazza	Initial transfer to OASIS template

Notes

[1] The CybOX Observable data model is actually defined in the CybOX Language, not in STIX.

[2] STIX does not inherently provide for marking at every level; an appropriate document selection language defined outside of STIX must be used (see Section **3.2**).

[3] XPath 1.0 is a language for selecting portions of XML documents.