THE MITRE CORPORATION

STIX™ 1.1.1
CORE SPECIFICATION (V1.1.1)

MAY 11, 2015

The Structured Threat Information eXpression (STIX™) framework defines eight 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 Core data model, which defines the STIX Package, the root object for all STIX content.

Acknowledgements

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

Trademark Information

STIX, the STIX logo, and CybOX are trademarks of The MITRE Corporation. All other trademarks are the property of their respective owners.

Warnings

MITRE PROVIDES STIX "AS IS" AND MAKES NO WARRANTY, EXPRESS OR IMPLIED, AS TO THE ACCURACY, CAPABILITY, EFFICIENCY, MERCHANTABILITY, OR FUNCTIONING OF STIX. IN NO EVENT WILL MITRE BE LIABLE FOR ANY GENERAL, CONSEQUENTIAL, INDIRECT, INCIDENTAL, EXEMPLARY, OR SPECIAL DAMAGES, RELATED TO STIX OR ANY DERIVATIVE THEREOF, WHETHER SUCH CLAIM IS BASED ON WARRANTY, CONTRACT, OR TORT, EVEN IF MITRE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.[1]

Feedback

The STIX development team welcomes any feedback regarding this document. Please send any comments, questions, or suggestions to stix@mitre.org.[2]

---

[1] For detailed information see [TOU].
[2] For more information about the STIX Language, please visit [STIX].

Table of Contents

1 Introduction

The Structured Threat Information eXpression (STIX^{TM}) framework defines eight top-level component data models: Observable[3], Indicator, Incident, TTP, ExploitTarget, CourseOfAction, Campaign, and ThreatActor. In addition, it defines a core data model for packaging and conveying content from any of these top-level components. This document serves as the specification for the STIX Core Version 1.1.1 data model, which is the unifying data model for all STIX content.

The STIX Core data model defines the concept of a STIX Package, the top-level object that is used to aggregate and convey all other objects of the STIX data models. The STIX Package has two main parts: a set of instances of each of the eight top-level components, which is the content of the STIX Package, and a STIX header, which can provide context for that content.

In Section 1.1 we list additional specification documents, and we provide document conventions in Section 1.2. 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 Core data model specification details in Section 3. References are provided in the final section.

1.1 STIX 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 specification overview document provides a comprehensive overview of the full set of STIX data models [STIX$_O$], which in addition to the eight 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$_O$] 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 (default vocabularies, data marking, and extensions), and the color white indicates the component data models. The Observable component data model is shown as an oval shape to indicate that it is defined as a CybOX specification (see [STIX$_O$] for details). This STIX Core specification document

---

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

is highlighted in its associated color (see Section 1.2.4.3). For a list of all STIX documents and related information sources, please see [STIX$_O$].
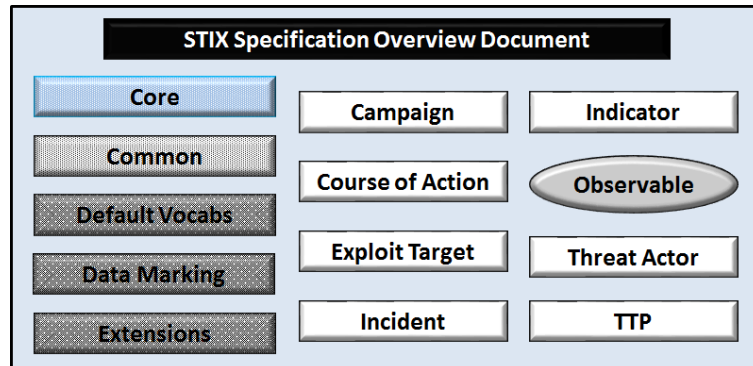

Figure 1-1. STIX Language v1.1.1 specification documents

All specification documents can be found on this STIX Website [STIX-SPECS].

1.2   Document Conventions

The following conventions are used in this document.

1.2.1   Key Words

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 *RFC 2119* [RFC2119].

1.2.2   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 the STIX Specification Overview [STIX$_O$].

  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*

1.2.3 UML Package References

Each STIX data model is captured in a different UML package (e.g., Core package, Campaign package, etc.). 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. The STIX[TM] 1.1.1 Specification Overview document [STIX$_O$] contains a list of the packages used by the Core 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 Core data model.

1.2.4 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. The fully described class can usually be found in a related diagram. 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.

1.2.4.1 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.

1.2.4.2 Diagram Icons and Arrow Types

Diagram icons are used in a UML diagram to indicate whether a shape is a class, enumeration or 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.

1.2.4.3 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 Core specification are illustrated in Figure 1-2. The overarching Core and Common data models, use the same light blue color coding.

Figure 1-2. Data model color coding

1.2.5   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

4

occurrences of the property, and a description column that describes the property. Package prefixes are provided for classes outside of the Core data model (see Section 1.2.3).

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.

1.2.6 Property and Class Descriptions

Each class and property defined in STIX is described using the format, "The X property <u>verb</u> Y." For example, in the specification for the STIX Indicator, we write, "The id property <u>specifies</u> a globally unique identifier for the kill chain 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
<u>captures</u>	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 <u>captured</u> include identitifying characteristics, time-related attributes, and a list of the tools used to collect the information. The Description property <u>captures</u> 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.
	Examples: 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.

6

2 Background Information

In this section, we provide high level information about the Core data model that is necessary to fully understand the specification details given in Section 3.

As will be explicitly detailed in Section 3, the STIX Core data model leverages all eight top-level component data models. Figure 2-1 illustrates the concept of a STIX Package, which acts as an *envelope* for the other top-level constructs in a STIX document. As stated in Section 1.1, each of these components is defined in a separate specification document.

Figure 2-1. A STIX Package

Because a STIX Package is simply a container, there is no requirement that the construct instances it contains are related in any way. In addition to the set of instances, a STIX Package contains a STIX Header, which characterizes general information such as a title, description and package intent. This information could be used to give context to the collection of objects contained in the package, if such context exists.

2.1   Component Data Models

Individual component data models define objects specific to each top-level STIX component construct: Observable; Indicator; Incident; Tactics, Techniques, and Procedures (TTPs); Exploit Target; Course of Action (COA); Campaign; and Threat Actor. These data models each provide the capability to fully express information about their targeted conceptual area. In the STIX framework, they are all optional and may be used separately or in concert, as appropriate, using whichever components and architectural relationships that are relevant for a given use case.

In the subsections below, a brief description is given for each component data model as well as a reference to the data model's individual specification document.

7

2.1.1 Observable

A STIX Observable (as defined with the CybOX Language[4]) represents stateful properties or measurable events pertinent to the operation of computers and networks. Implicit in this is a practical need for descriptive capability of two forms of observables: "observable instances" and "observable patterns." Observable instances represent actual specific observations that took place in the cyber domain. The property details of this observation are specific and unambiguous. Observable patterns represent conditions for a potential observation that may occur in the future or may have already occurred and exists in a body of observable instances. These conditions may be anything from very specific concrete patterns that would match very specific observable instances to more abstract generalized patterns that have the potential to match against a broad range of potential observable instances.

2.1.2 Indicator

A STIX Indicator conveys specific Observable patterns combined with contextual information intended to represent artifacts and/or behaviors of interest within a cyber security context. Please see the STIX Indicator data model specification [STIX$_{IND}$] for details.

2.1.3 Incident

A STIX Incident corresponds to sets of related security events affecting an organization, along with information discovered or decided during an incident response investigation. Please see the STIX Incident data model specification [STIX$_{INC}$] for details.

2.1.4 Tactics, Techniques and Procedures (TTP)

A STIX Tactics, Techniques, and Procedures (TTP) is used to represent the behavior or modus operandi of cyber adversaries. Please see the STIX TTP data model specification [STIX$_{TTP}$] for details.

2.1.5 Campaign

A STIX Campaign represents a set of TTPs, Incidents, or Threat Actors that together express a common intent or desired effect. Please see the STIX Campaign data model specification [STIX$_{CAM}$] for details.

2.1.6 Threat Actor

A STIX Threat Actor is a characterization of a malicious actor (or adversary) representing a cyber attack threat including presumed intent and historically observed behavior. Please see the STIX Threat Actor data model specification [STIX$_{TA}$] for details.

---

[4] CybOX specification documents will be created after STIX specification documents are completed.

2.1.7 Exploit Target

A STIX Exploit Target conveys information about a vulnerability, weakness, or misconfiguration in software, systems, networks, or configurations that may be targeted for exploitation by an adversary. Please see the STIX Exploit Target data model specification [STIX$_{ET}$] for details.

2.1.8 Course of Action (COA)

A STIX Course of Action (COA) is used to convey information about courses of action that may be taken either in response to an attack or as a preventative measure prior to an attack. Please see the STIX Course of Action data model specification [STIX$_{COA}$] for details.

3   STIX Core Data Model

The primary class of the STIX Core package is the `STIXType` class, which defines a bundle of information characterized in the Structured Threat Information eXpression (STIX) language. We refer to this bundle of information as a "STIX Package."[5]

The properties of the `STIXType` class, are illustrated in the UML diagram given in Figure 3-1.

---

[5] Throughout this section, a "STIX Package" denotes an object of type `STIXType` class.
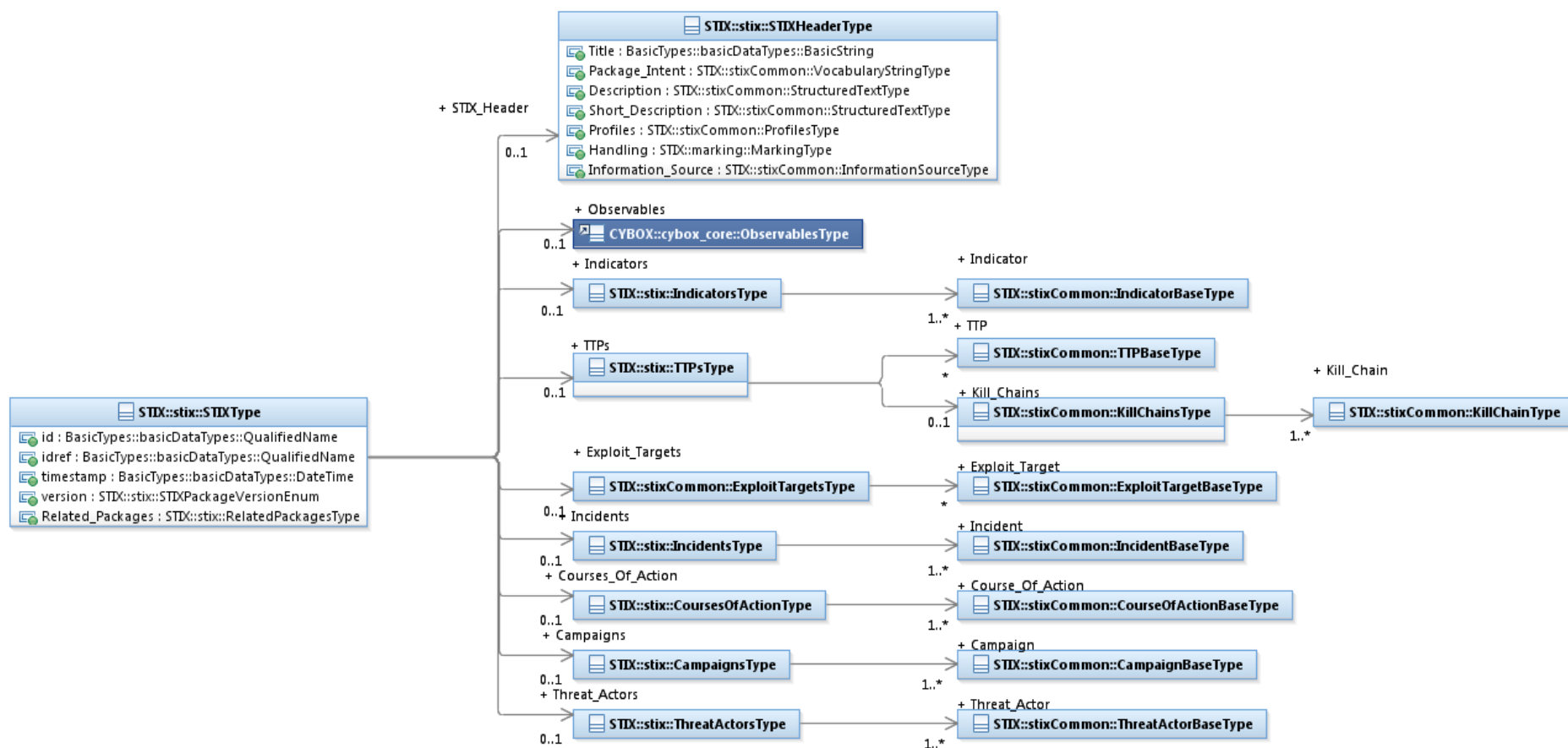
Figure 3-1. UML diagram of the `STIXType` class

Table 3-1. Properties of The `STIXType` class

Name	Type	Multiplicity	Description
id	`BasicDataTypes: QualifiedName`	0..1	The `id` property specifies a globally unique identifier for the STIX Package.
idref	`BasicDataTypes: QualifiedName`	0..1	The `idref` property specifies an identifier reference to a STIX Package specified elsewhere. When the `idref` property is used, the `id` property MUST NOT also be specified and the other properties of the `STIXType` class SHOULD NOT hold any content.
timestamp	`BasicDataTypes: DateTime`	0..1	The `timestamp` property specifies a timestamp for the definition of a specific version of a STIX Package. When used in conjunction with the `id` property, this property specifies the definition time for the specific version of the STIX Package. When used in conjunction with the `idref` property, this property specifies a reference to a specific version of a STIX Package defined elsewhere. This property has no defined semantic meaning if used in the absence of either the `id` or `idref` properties.
version	`STIXPackageVersionEnum`	0..1	The `version` property specifies the version identifier of the STIX Core data model used to capture the information associated with the STIX Package.
STIX_Header	`STIXHeaderType`	0..1	The `STIX_Header` property characterizes the metadata for this package of STIX content.
Observables	`cybox:ObservablesType`	0..1	The `Observables` property specifies a set of one or more cyber observables.
Indicators	`IndicatorsType`	0..1	The `Indicators` property specifies a set of one or more cyber threat Indicators.
TTPs	`TTPsType`	0..1	The `TTP` property specifies a set of one or more cyber threat adversary Tactics, Techniques or Procedures (TTPs), including Kill Chains.

Exploit_Targets	stixCommon:ExploitTargetsType	0..1	The Exploit_Targets property specifies a set of one or more potential targets for exploitation.
Incidents	IncidentsType	0..1	The Incidents property specifies a set of one or more cyber threat Incidents.
Courses_Of_Action	CoursesOfActionType	0..1	The CoursesOfActions property specifies a set of one or more Courses of Action that could be taken in regard to one of more cyber threats.
Campaigns	CampaignsType	0..1	The Campaigns property specifies a set of one or more Campaigns.
Threat_Actors	ThreatActorsType	0..1	The ThreatActors property specifies a set of one or more Threat Actors.
Related_Packages	RelatedPackagesType	0..1	The Related_Packages property specifies a set of one or more Packages which may be relevant to this STIX Package.

3.1 STIXPackageVersionType Enumeration

The STIXPackageVersionType enumeration is an inventory of all possible versions of the STIX Core data model. The version used in the STIX Package corresponds to the version of STIX in use. The enumeration literals are given in Table 3-2.

Table 3-2. Literals of the `STIXPackageVersionType` enumeration

Enumeration Literal	Description
1.0	STIX Core data model Version 1.0
1.0.1	STIX Core data model Version 1.0.1
1.1	STIX Core data model Version 1.1
1.1.1	STIX Core data model Version 1.1.1

3.2   STIXHeaderType Class

The `STIXHeaderType` class provides a structure for characterizing a package of STIX content.

The properties of the `STIXHeaderType` class are given in Table 3-3.

Table 3-3. Properties of the `STIXHeaderType` class

Name	Type	Multiplicity	Description
Title	`basicDataTypes:BasicString`	0..1	The `Title` property captures a title for the STIX Package and reflects what the content producer thinks the Package as a whole should be called. The `Title` property is typically used by humans to reference a particular Package; however, it is not suggested for correlation.
Package_Intent	`stixCommon: VocabularyStringType`	0..*	The `Package_Intent` property specifies the intended purpose(s) or use(s) for The STIX Package. Examples of potential purposes are *phishing*, *exploit characterization* and *malware samples* (these

			specific values are only provided to help explain the property: they are neither recommended types nor necessarily part of any existing vocabulary). The content creator may choose any arbitrary value or may constrain the set of possible values by referencing an externally-defined vocabulary or leveraging a formally defined vocabulary extending from the `stixCommon:ControlledVocabularyStringType` class. The STIX default vocabulary class for use in this property is *'PackageIntentVocab-1.0'*.
Description	`stixCommon:StructuredTextType`	0..1	The `Description` property captures a textual description of the STIX Package. Any length is permitted. Optional formatting is supported via the `structuring_format` property of the `StructuredTextType` class.
Short_Description	`stixCommon:StructuredTextType`	0..1	The `Short_Description` property captures a short textual description of the STIX Package. This property is secondary and should only be used if the `Description` property is already populated and another, shorter description is available.
Profiles	`stixCommon:ProfilesType`	0..1	The `Profiles` property specifies a set of one or more profiles that the content of the STIX Package conforms to.
Handling	`marking:MarkingType`	0..1	The `Handling` property specifies the appropriate data handling markings for the properties of this STIX Package. The marking scope is limited to the STIX Package and the content it contains. Note that data handling markings can also be specified at a higher level.
Information_Source	`stixCommon:InformationSourceType`	0..1	The `Information_Source` property characterizes the source of the STIX Package and all of its contained information. Examples of details captured include identitifying characteristics, time-related attributes, and a list of the tools used to collect the information.

3.3   Content Aggregation Types

Each component type has an associated aggregation class that has one main property – a set of one or more instances of that component type. The aggregation class for Observables, `cybox_core:ObservablesType`, is defined in [CybOX_{COR}].

3.3.1   CampaignsType Class

The `CampaignsType` class specifies a set of one or more cyber threat Campaigns.

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

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

Name	Type	Multiplicity	Description
Campaign	`stixCommon:CampaignBaseType`	1..*	The `Campaign` property characterizes a cyber threat Campaign. The `stixCommon:CampaignBaseType` class is a minimal base class that is intended to be extended. The default and strongly recommended class to fully implement a Campaign is the `campaign:CampaignType` class defined in [STIX_{CAM}]. Base classes are used to minimize interdependence between STIX components, not to enable or encourage conflicting syntactic variation. However, through the use of the `idref` property, a reference to a Campaign defined elsewhere can be specified via the direct use of the `stixCommon:CampaignBaseType` class.

3.3.2   CoursesOfActionType Class

The `CoursesOfActionType` class specifies a set of one or more actions that could be taken in regard to cyber threats.

The properties of the `CoursesOfActionType` class are given in Table 3-5.

Table 3-5. Properties of the `CoursesOfActionType` class

Name	Type	Multiplicity	Description
Course_Of_Action	`stixCommon: CourseOfActionBaseType`	1..*	The `Course_Of_Action` property characterizes a Course of Action that could be taken in regard to one of more cyber threats. The `stixCommon:CourseOfActionBaseType` class is a minimal base class that is intended to be extended. The default and strongly RECOMMENDED class to fully implement a Course of Action is the `coa:CourseOfActionType` class defined in [STIX_{COA}]. Base classes are used to minimize interdependence between STIX components, not to enable or encourage conflicting syntactic variation. However, through the use of the `idref` property, a reference to a Course of Action defined elsewhere can be specified via the direct use of the `stixCommon:CourseOfActionBaseType` class.

3.3.3   IncidentsType Class

The `IncidentsType` class specifies a set of one or more cyber threat Incidents.

The properties of the `IncidentsType` class are given in Table 3-6.

Table 3-6. Properties of the `IncidentsType` class

Name	Type	Multiplicity	Description
Incident	`stixCommon: IncidentBaseType`	1..*	The `Incident` property characterizes a cyber threat Incident. The `stixCommon:IncidentBaseType` class is a minimal base class that is intended to be extended. The default and strongly recommended class to fully implement an Incident is the `incident:IncidentType` class defined in [STIX_{INC}]. Base classes are used to minimize interdependence between STIX components, not to enable or encourage conflicting syntactic variation. However, through the use of the `idref` property, a reference to an Incident defined elsewhere can be specified via the direct use of the `stixCommon:IncidentBaseType` class.

17

3.3.4 IndicatorsType Class

The `IndicatorsType` class specifies a set of one or more cyber threat Indicators.

The properties of the `IndicatorsType` class are given in Table 3-7.

Table 3-7. Properties of the `IndicatorsType` class

Name	Type	Multiplicity	Description
Indicator	stixCommon: IndicatorBaseType	1..*	The `Indicator` property characterizes a cyber threat Indicator. The `stixCommon:IndicatorBaseType` class is a minimal base class that is intended to be extended. The default and strongly recommended class to fully implement an Indicator is the `indicator:IndicatorType` class defined in [STIX_{IND}]. Base classes are used to minimize interdependence between STIX components, not to enable or encourage conflicting syntactic variation. However, through the use of the `idref` property, a reference to an Indicator defined elsewhere can be specified via the direct use of the `stixCommon:IndicatorBaseType` class.

3.3.5 ThreatActorsType Class

The `ThreatActorsType` class specifies a set of one or more cyber Threat Actors.

The properties of the `ThreatActorsType` class are given in Table 3-8.

Table 3-8. Properties of the `ThreatActorsType` class

Name	Type	Multiplicity	Description
Threat_Actor	stixCommon: ThreatActorBaseType	1..*	The `ThreatActor` property characterizes a cyber Threat Actor. The `stixCommon:ThreatActorBaseType` class is a minimal base class that is intended to be extended. The default and strongly recommended class to fully

			implement an ThreatActor is the `ta:ThreatActorType` class defined in [STIX_{TA}]. Base classes are used to minimize interdependence between STIX components, not to enable or encourage conflicting syntactic variation. However, through the use of the `idref` property, a reference to a Threat Actor defined elsewhere can be specified via the direct use of the `stixCommon:ThreatActorBaseType` class.

3.3.6 TTPsType Class

The `TTPsType` class specifies a set of one or more cyber threat TTPs.

The properties of the `TTPsType` class are given in Table 3-9.

Table 3-9. Properties of the `TTPsType` class

Name	Type	Multiplicity	Description
TTP	`stixCommon:TTPBaseType`	0..*	The TTP property characterizes a cyber threat adversary Tactic, Technique or Procedure (TTP). The `stixCommon:TTPBaseType` class is a minimal base class that is intended to be extended. The default and strongly recommended class to fully implement a TTP is the `ttp:TTPType` class defined in [STIX_{TTP}]. Base classes are used to minimize interdependence between STIX components, not to enable or encourage conflicting syntactic variation. However, through the use of the `idref` property, a reference to a TTP defined elsewhere can be specified via the direct use of the `stixCommon:TTPBaseType` class.
Kill_Chains	`stixCommon:KillChainsType`	0..1	A cyber kill chain is a phase-based model to describe the stages of an attack. The `Kill_Chains` property specifies a set of one or more specific kill chain definitions. The `kill_chain` property is further defined in the STIX Common specification document. Note that kill chains may also be defined using the `Kill_Chains` property of the TTP `TTPType` class, which is equivalent to this property. Suggested practice is to use the TTP `TTPType` `Kill_Chains` property (rather than this property) to define a kill chain.

3.4   RelatedPackagesType Class

The `RelatedPackagesType` class specifies a set of one or more STIX Package related to this STIX Package. It extends the `GenericRelationShipListType` superclass defined in the STIX Common data model, which specifies the scope (whether the elements of the set are related individually or as a group).

The UML diagram corresponding to the `RelatedPackagesType` class is shown in Figure 3-2, and the specialized properties are shown in Table 3-10.
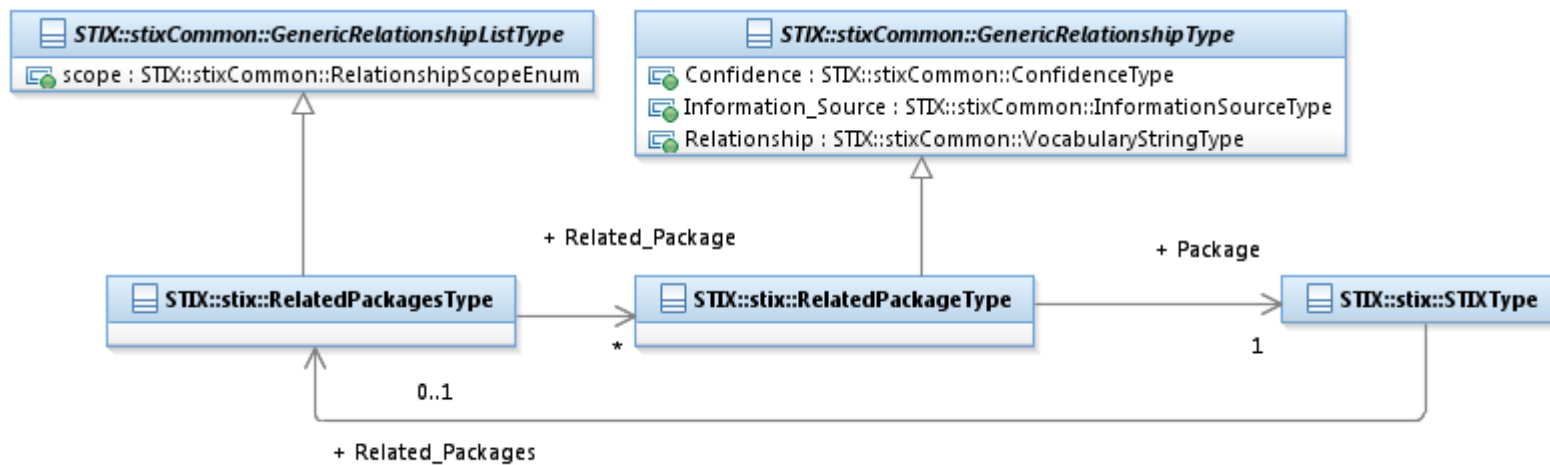

Figure 3-2. UML diagram for `RelatedPackagesType` class

In addition to being a property of the `STIXType` class, `Related_Packages` is a property of all of the top-level component types.

The property table given in Table 3-10 corresponds to the UML diagram shown in Figure 3-2.

Table 3-10. Properties of `RelatedPackagesType` class

Name	Type	Multiplicity	Description
Related_Package	`RelatedPackageType`	0..*	The `Related_Package` property characterizes a relationship to one or more other STIX Packages.

3.4.1 RelatedPackageType Class

The `RelatedPackageType` class identifies or characterizes the relationship of STIX Package to another.

Table 3-11. Properties of `RelatedPackageType` class

Name	Type	Multiplicity	Description
Package	`STIXType`	1	The `Package` property captures or references a STIX Package related to this STIX Package.

References

References made in this document are listed below.

[CybOX_COR]   CybOX Core Specification (*not yet available*).

[REL]   STIX^TM 1.1.1 Indicator Model as implement in XSD
https://stix.mitre.org/language/version4.1/xxx_schema.xsd

[RFC2119]   RFC 2119 – Key words for use in RFCs to Indicate Requirement Levels
http://www.ietf.org/rfc/rfc2119.txt

[STIX]   STIX^TM Web Site
http://stix.mitre.org

[STIX-SPECS]   STIX^TM Project Github Site
http://github.com/STIXProject/specifications

[STIX_CAM]   STIX^TM 1.1.1 Campaign Specification (v1.1.1)
http://stix.mitre.org/about/documents/XXXX.pdf

[STIX_COA]   STIX^TM 1.1.1 Course of Action (COA) Specification (v1.1.1)
http://stix.mitre.org/about/documents/XXXX.pdf

[STIX_COM]   STIX^TM 1.1.1 Common Specification (v1.1.1)
http://stix.mitre.org/about/documents/XXXX.pdf

[STIX_ET]   STIX^TM 1.1.1 Exploit Target Specification (v1.1.1)
http://stix.mitre.org/about/documents/XXXX.pdf

[STIX_INC]   STIX^TM 1.1.1 Incident Specification (v1.1.1)
http://stix.mitre.org/about/documents/XXXX.pdf

[STIX_IND]   STIX^TM 1.1.1 Indicator Specification (v2.1.1)
http://stix.mitre.org/about/documents/XXXX.pdf

[STIX_O]   STIX^TM 1.1.1 Specification Overview
http://stix.mitre.org/about/documents/XXXX.pdf

[STIX_TA]   STIX^TM 1.1.1 Threat Actor Specification (v1.1.1)
http://stix.mitre.org/about/documents/XXXX.pdf

[STIX_TTP]   STIX^TM 1.1.1 TTP Specification (v1.1.1)

http://stix.mitre.org/about/documents/XXXX.pdf

[TOU]   Terms of Use
http://stix.mitre.org/about/termsofuse.html