CybOXTM Version 2.1.1 Part 4: Default Extensions

Working Draft 01

15 December 2015

Technical Committee:

[OASIS Cyber Threat Intelligence (CTI) TC](https://www.oasis-open.org/committees/cti/)

Chair:

Richard Struse ([Richard.Struse@HQ.DHS.GOV](mailto:Richard.Struse@HQ.DHS.GOV)), [DHS Office of Cybersecurity and Communications (CS&C)](http://www.dhs.gov/office-cybersecurity-and-communications)

Editors:

Desiree Beck ([dbeck@mitre.org](mailto:ikirillov@mitre.org)), [MITRE Corporation](http://www.mitre.org/)

Trey Darley ([trey@soltra.com](mailto:trey@soltra.com)), [Soltra](http://www.soltra.com/)

Ivan Kirillov ([ikirillov@mitre.org](mailto:ikirillov@mitre.org)), [MITRE Corporation](http://www.mitre.org/)

Rich Piazza ([rpiazza@mitre.org](mailto:ikirillov@mitre.org)), [MITRE Corporation](http://www.mitre.org/)

Additional artifacts:

This prose specification is one component of a Work Product, which consists of:

* *CybOX™ Version 2.1.1 Part 01: Overview*.
* *CybOX™ Version 2.1.1 Part 02: Common*. [URI]
* *CybOX™ Version 2.1.1 Part 03: Core*. [URI]
* *CybOX™ Version 2.1.1 Part 04: Default Extensions*. (this document)
* *CybOX™ Version 2.1.1 Part 05: Default Vocabularies*. [URI]
* *CybOX™ Version 2.1.1 Part 06: UML Model*. [URI]
* *CybOX™ Version 2.1.1 Part 07: API Object*. [URI]
* *CybOX™ Version 2.1.1 Part 08: ARP Cache Object*. [URI]
* *CybOX™ Version 2.1.1 Part 09: AS Object*. [URI]
* *CybOXTM Version 2.1.1 Part 10: Account Object. [URI]*
* *CybOXTM Version 2.1.1 Part 11: Address Object. [URI]*
* *CybOXTM Version 2.1.1 Part 12: Archive File Object. [URI]*
* *CybOXTM Version 2.1.1 Part 13: Artifact Object. [URI]*
* *CybOXTM Version 2.1.1 Part 14: Code Object. [URI]*
* *CybOXTM Version 2.1.1 Part 15: Custom Object. [URI]*
* *CybOXTM Version 2.1.1 Part 16: DNS Cache Object. [URI]*
* *CybOXTM Version 2.1.1 Part 17: DNS Query Object. [URI]*
* *CybOXTM Version 2.1.1 Part 18: DNS Record Object. [URI]*
* *CybOXTM Version 2.1.1 Part 19: Device Object. [URI]*
* *CybOXTM Version 2.1.1 Part 20: Disk Object. [URI]*
* *CybOXTM Version 2.1.1 Part 21: Disk Partition Object. [URI]*
* *CybOXTM Version 2.1.1 Part 22: Domain Name Object. [URI]*
* *CybOXTM Version 2.1.1 Part 23: Email Message Object. [URI]*
* *CybOXTM Version 2.1.1 Part 24: File Object. [URI]*
* *CybOXTM Version 2.1.1 Part 25: GUI Dialogbox Object. [URI]*
* *CybOXTM Version 2.1.1 Part 26: GUI Object. [URI]*
* *CybOXTM Version 2.1.1 Part 27: GUI Window Object. [URI]*
* *CybOXTM Version 2.1.1 Part 28: HTTP Session Object. [URI]*
* *CybOXTM Version 2.1.1 Part 29: Hostname Object. [URI]*
* *CybOXTM Version 2.1.1 Part 30: Image File Object. [URI]*
* *CybOXTM Version 2.1.1 Part 31: Library File Object. [URI]*
* *CybOXTM Version 2.1.1 Part 32: Link Object. [URI]*
* *CybOXTM Version 2.1.1 Part 33: Linux Package Object. [URI]*
* *CybOXTM Version 2.1.1 Part 34: Memory Object. [URI]*
* *CybOXTM Version 2.1.1 Part 35: Mutex Object. [URI]*
* *CybOXTM Version 2.1.1 Part 36: Network Connection Object. [URI]*
* *CybOXTM Version 2.1.1 Part 37: Network Flow Object. [URI]*
* *CybOXTM Version 2.1.1 Part 38: Network Packet Object. [URI]*
* *CybOXTM Version 2.1.1 Part 39: Network Route Entry Object. [URI]*
* *CybOXTM Version 2.1.1 Part 40: Network Route Object. [URI]*
* *CybOXTM Version 2.1.1 Part 41: Network Socket Object. [URI]*
* *CybOXTM Version 2.1.1 Part 42: Network Subnet Object. [URI]*
* *CybOXTM Version 2.1.1 Part 43: PDF File Object. [URI]*
* *CybOXTM Version 2.1.1 Part 44: Pipe Object. [URI]*
* *CybOXTM Version 2.1.1 Part 45: Port Object. [URI]*
* *CybOXTM Version 2.1.1 Part 46: Process Object. [URI]*
* *CybOXTM Version 2.1.1 Part 47: Product Object. [URI]*
* *CybOXTM Version 2.1.1 Part 48: SMS Message Object. [URI]*
* *CybOXTM Version 2.1.1 Part 49: Semaphore Object. [URI]*
* *CybOXTM Version 2.1.1 Part 50: Socket Address Object. [URI]*
* *CybOXTM Version 2.1.1 Part 51: System Object. [URI]*
* *CybOXTM Version 2.1.1 Part 52: URI Object. [URI]*
* *CybOXTM Version 2.1.1 Part 53: URL History Object. [URI]*
* *CybOXTM Version 2.1.1 Part 54: Unix File Object. [URI]*
* *CybOXTM Version 2.1.1 Part 55: Unix Network Route Entry Object. [URI]*
* *CybOXTM Version 2.1.1 Part 56: Unix Pipe Object. [URI]*
* *CybOXTM Version 2.1.1 Part 57: Unix Process Object. [URI]*
* *CybOXTM Version 2.1.1 Part 58: Unix User Account Object. [URI]*
* *CybOXTM Version 2.1.1 Part 59: Unix Volume Object. [URI]*
* *CybOXTM Version 2.1.1 Part 60: User Account Object. [URI]*
* *CybOXTM Version 2.1.1 Part 61: User Session Object. [URI]*
* *CybOXTM Version 2.1.1 Part 62: Volume Object. [URI]*
* *CybOXTM Version 2.1.1 Part 63: Whois Object. [URI]*
* *CybOXTM Version 2.1.1 Part 64: Win Computer Account Object. [URI]*
* *CybOXTM Version 2.1.1 Part 65: Win Critical Section Object. [URI]*
* *CybOXTM Version 2.1.1 Part 66: Win Driver Object. [URI]*
* *CybOXTM Version 2.1.1 Part 67: Win Event Log Object. [URI]*
* *CybOXTM Version 2.1.1 Part 68: Win Event Object. [URI]*
* *CybOXTM Version 2.1.1 Part 69: Win Executable File Object. [URI]*
* *CybOXTM Version 2.1.1 Part 70: Win File Object. [URI]*
* *CybOXTM Version 2.1.1 Part 71: Win Filemapping Object. [URI]*
* *CybOXTM Version 2.1.1 Part 72: Win Handle Object. [URI]*
* *CybOXTM Version 2.1.1 Part 73: Win Hook Object. [URI]*
* *CybOXTM Version 2.1.1 Part 74: Win Kernel Hook Object. [URI]*
* *CybOXTM Version 2.1.1 Part 75: Win Kernel Object. [URI]*
* *CybOXTM Version 2.1.1 Part 76: Win Mailslot Object. [URI]*
* *CybOXTM Version 2.1.1 Part 77: Win Memory Page Region Object. [URI]*
* *CybOXTM Version 2.1.1 Part 78: Win Mutex Object. [URI]*
* *CybOXTM Version 2.1.1 Part 79: Win Network Route Entry Object. [URI]*
* *CybOXTM Version 2.1.1 Part 80: Win Network Share Object. [URI]*
* *CybOXTM Version 2.1.1 Part 81: Win Pipe Object. [URI]*
* *CybOXTM Version 2.1.1 Part 82: Win Prefetch Object. [URI]*
* *CybOXTM Version 2.1.1 Part 83: Win Process Object. [URI]*
* *CybOXTM Version 2.1.1 Part 84: Win Registry Key Object. [URI]*
* *CybOXTM Version 2.1.1 Part 85: Win Semaphore Object. [URI]*
* *CybOXTM Version 2.1.1 Part 86: Win Service Object. [URI]*
* *CybOXTM Version 2.1.1 Part 87: Win System Object. [URI]*
* *CybOXTM Version 2.1.1 Part 88: Win System Restore Object. [URI]*
* *CybOXTM Version 2.1.1 Part 89: Win Task Object. [URI]*
* *CybOXTM Version 2.1.1 Part 90: Win Thread Object. [URI]*
* *CybOXTM Version 2.1.1 Part 91: Win User Account Object. [URI]*
* *CybOXTM Version 2.1.1 Part 92: Win Volume Object. [URI]*
* *CybOXTM Version 2.1.1 Part 93: Win Waitable Timer Object. [URI]*
* *CybOXTM Version 2.1.1 Part 94: X509 Certificate Object. [URI]*

Related work:

This specification is related to:

* *STIXTM Version 1.2.1 (placeholder)*

Abstract:

The Cyber Observable Expression (CybOX™) is a standardized language for encoding and communicating high-fidelity information about cyber observables, whether dynamic events or stateful measures that are observable in the operational cyber domain. By specifying a common structured schematic mechanism for these cyber observables, the intent is to enable the potential for detailed automatable sharing, mapping, detection, and analysis heuristics. This specification document defines the default extensions, data model, which enables external data models to be included in CybOX content.

Status:

This [Working Draft](https://www.oasis-open.org/policies-guidelines/tc-process#dWorkingDraft) (WD) has been produced by one or more TC Members; it has not yet been voted on by the TC or [approved](https://www.oasis-open.org/policies-guidelines/tc-process#committeeDraft) as a Committee Draft (Committee Specification Draft or a Committee Note Draft). The OASIS document [Approval Process](https://www.oasis-open.org/policies-guidelines/tc-process#standApprovProcess) begins officially with a TC vote to approve a WD as a Committee Draft. A TC may approve a Working Draft, revise it, and re-approve it any number of times as a Committee Draft.

URI patterns:

Initial publication URI:  
http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part12-extensions/stix-v1.2.1-csd01-part12-extensions.docx

Permanent “Latest version” URI:  
http://docs.oasis-open.org/cti/stix/v1.2.1/stix-v1.2.1-part12-extensions.docx

(Managed by OASIS TC Administration; please don’t modify.)

Copyright © OASIS Open 2016. All Rights Reserved.

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](https://www.oasis-open.org/policies-guidelines/ipr) 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.

Portions copyright © United States Government 2012-2016. All Rights Reserved. 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 6](#_Toc440957888)

[1.1 CybOXTM Specification Documents 6](#_Toc440957889)

[1.2 Document Conventions 6](#_Toc440957890)

[1.2.1 Fonts 6](#_Toc440957891)

[1.2.2 UML Package References 7](#_Toc440957892)

[1.2.3 UML Diagrams 7](#_Toc440957893)

[1.2.3.1 Class Properties 7](#_Toc440957894)

[1.2.3.2 Diagram Icons and Arrow Types 7](#_Toc440957895)

[1.2.4 Property Table Notation 8](#_Toc440957896)

[1.2.5 Property and Class Descriptions 8](#_Toc440957897)

[1.3 Terminology 9](#_Toc440957898)

[1.4 Normative References 9](#_Toc440957899)

[2 Background Information 10](#_Toc440957900)

[2.1 Extending CybOX 10](#_Toc440957901)

[3 CybOX Default Extension Data Models 12](#_Toc440957902)

[3.1 Addresses: Location Data Model 12](#_Toc440957903)

[3.1.1 CIQAddress3.0InstanceType Class 12](#_Toc440957904)

[3.2 Platform Data Model 13](#_Toc440957905)

[3.2.1 CPE23PlatformSpecificationType Class 13](#_Toc440957906)

[4 Conformance 14](#_Toc440957907)

[Appendix A. Acknowledgments 15](#_Toc440957908)

[Appendix B. Revision History 17](#_Toc440957909)

# Introduction

[All text is normative unless otherwise labeled.]

The Cyber Observable Expression (CybOX™) Language provides a common structure for representing cyber observables across and among the operational areas of enterprise cyber security. CybOX improves the consistency, efficiency, and interoperability of deployed tools and processes, and it increases overall situational awareness by enabling the potential for detailed automatable sharing, mapping, detection, and analysis heuristics.

This document serves as the specification for the CybOX Default Extensions Version 2.1.1 data model, which enables external data models to be included in CybOX content.

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**. In Section **2**, we give background information necessary to fully understand the Default Extensions data model. We present the Default Extensions data model specification details in Section **3** and conformance information in Section **4**.

## CybOXTM Specification Documents

The CybOX 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 individual data models that compose the full CybOX UML model.

CybOX has a modular design comprising two fundamental data models and a collection of Object data models. The fundamental data models – CybOX Core and CybOX Common – provide essential CybOX structure and functionality. The CybOX Objects, defined in individual data models, are precise characterizations of particular types of observable cyber entities (e.g., HTTP session, Windows registry key, DNS query).

Use of the CybOX Core and Common data models is required; however, use of the CybOX Object data models is purely optional: users select and use only those Objects and corresponding data models that are needed. Importing the entire [CybOX suite of data models](#AdditionalArtifacts) is not necessary.

The [*CybOX™ Version 2.1.1 Part 1: Overview*](#AdditionalArtifacts) document provides a comprehensive overview of the full set of CybOX data models, which in addition to the Core, Common, and numerous Object data models, includes various extension data models and a vocabularies data model, which contains a set of default controlled vocabularies. [*CybOX™ Version 2.1.1 Part 1: Overview*](#AdditionalArtifacts) also summarizes the relationship of CybOX to other externally defined data models, and outlines general CybOX data model conventions.

## Document Conventions

The following conventions are used in this document.

### Fonts

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

* Capitalization is used for CybOX high-level concepts, which are defined in [*CybOX™ Version 2.1.1 Part 1: Overview*](#AdditionalArtifacts).

Examples: Action, Object, Event, Property

* The Courier New font is used for writing UML objects.

Examples: ActionType, cyboxCommon:BaseObjectPropertyType

Note that all high-level concepts have a corresponding UML object. For example, the Action high-level concept is associated with a UML class named, ActionType.

* The ‘*italic’* font (withsingle quotes) is used for noting actual, explicit values for CybOX Language properties. The *italic* font (without quotes) is used for noting example values.

Example:  *‘HashNameVocab-1.0,’ high, medium, low*

### UML Package References

Each CybOX data model is captured in a different UML package (e.g., Core package) where the packages together compose the full [CybOX UML model](#AdditionalArtifacts). 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.

### UML Diagrams

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

#### 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. In particular, we will always capture properties of UML data types as attributes.

#### 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 ‑. 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. |

### 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 Default Extensions 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.

### Property and Class Descriptions

Each class and property defined in CybOX is described using the format, “The X property verbY.” For example, in the specification for the CybOX Core data model, we write, “The id property specifies a globally unique identifier for the Action.” 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 CybOX.

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

|  |  |
| --- | --- |
| **Verb** | **CybOX 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 Observable\_Source property characterizes the source of the Observable information. Examples of details captured include identifying characteristics, time-related attributes, and a list of the tools used to collect the information.  The Description property captures a textual description of the Action. |
| 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 Action property characterizes a cyber observable Action.  The Obfuscation\_Technique property characterizes a technique an attacker could potentially leverage to obfuscate the Observable. |
| 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 cybox\_major\_version property specifies the major version of the CybOX language used for the set of Observables. |

## 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].

## 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>.

[CPE] Common Platform Enumeration (CPE). (2014, Nov. 28). The MITRE Corporation. [Online]. Available: <http://cpe.mitre.org>.

**[CIQ]** *Customer Information Quality (CIQ) Specifications Version 3.0*. Edited by Ram Kumar. 8 April 2008. OASIS Public Review Draft 03. Available: <http://docs.oasis-open.org/ciq/v3.0/specs/ciq-specs-v3.html>.

# Background Information

In this section, we provide high-level information that is necessary to fully understand the Default Extension data models specification details given in Section **3**.

## Extending CybOX

In any UML model, an arbitrary class can usually be extended, but in general, extending a data model is antithetical to the concept behind a standardized data model used for sharing information. However, some of the concepts that need to be represented in CybOX already are defined in established data models outside of CybOX. To support the inclusion of those data models into CybOX, a number of extension point classes have been identified. The number of extension points is not fixed, and others might be added in the future, if the need arises.

This document defines the default extension data models and their associated classes, which are specializations of the extension point classes. These default extension classes compose the currently available extension data models. The extensions defined in this document are defaults; others can be used. Note that some extension point classes do not have a corresponding default data model externally defined. Additionally, some extension point classes have no corresponding extension class defined in the CybOX Default Extension data models (see the extension point class definitions for more details).

**Table 2‑1** shows the relationship between the extension point classes and the default extension classes.

Table ‑. Extension points classes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Extension Point Class** | **Abstract?** | **Contains Properties?** | **Externally Defined Data Model?** | **Default Extension Classes** |
| cyboxCore: DomainSpecificObjectPropertiesType | Y | N | N | *None* |
| cyboxCommon:LocationType | N | Y | Y | ciq\_address\_3.0:  CIQAddress3.0InstanceType |
| cyboxCommon:  PlatformSpecificationType | N | Y | Y | cpe\_2.3:  CPE23PlatformSpecificationType |
| cyboxCommon:  ToolSpecificDataType | Y | N | N | *None* |

From a UML package perspective, **Table 2‑2** shows the relationships between the various UML packages that exist to support a modular approach to creating extensions to the CybOX data models. Each extension data model has its own package. The primary class of each of those packages specializes an extension point class that is contained in one of the other packages of the CybOX model. The extension classes generally have one or more properties to support the connection between the CybOX and the externally defined data models. Those properties are either associated with a class from the corresponding external package or contain a text specification in the native format of the external data model. In the former case, we provide the name of the external defined package in the table. If a text specification is used, then the package name is not applicable, because there is no formally defined UML package.

Table ‑. Packages Associated with the Default Extension Data Models

|  |  |  |
| --- | --- | --- |
| **Extension Class Package** | **Extension Point Class Package** | **External Data Model Package** |
| ciq\_address\_3.0 | cyboxCommon | a |
| cpe2.3 | cyboxCommon | cpe |

# 

# CybOX Default Extension Data Models

Each CybOX extension data model contains a primary class, called the extension class that extends a class in one or more other CybOX data models. In sections **3.1** and **3.2** we define the classes of each extension data model. Externally defined data models are contained in a UML package named external. The names of the packages used in this document for the external data models are often aliases (e.g., the package a is an alias for urn:oasis:names:tc:ciq:xal from the external data model).

## Addresses: Location Data Model

The default extension class for expressing geographic address information in CybOX 2.2.1 is the CIQAddress3.0InstanceType class. The underlying data model being referenced is the structured characterization of addresses of the OASIS Customer Information Quality (CIQ) Specification as defined in [**[CIQ]**](#ciq).

### CIQAddress3.0InstanceType Class

The CIQAddress3.0InstanceType class is defined as a subclass to extend the cybox\_common:LocationType superclass and belongs to the ciq\_address\_3.0 package. As shown in **Figure 3‑1**, the CIQAddress3.0InstanceType class imports and leverages version 3.0 of the OASIS CIQ-PIL schema for structured characterization of addresses.

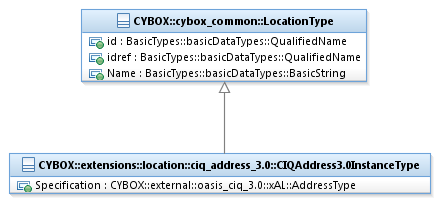


Figure ‑. UML diagram of the CIQAddress3.0InstanceType class

The property table for the CIQAddress3.0InstanceType class is given in **Table 3‑1**.

Table ‑. Properties of the CIQAddress3.0InstanceType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **Specification** | a:AddressType | 1 | The Specification property specifies a potentially long set of address-related information including address type (e.g., business, rural)*,* country, administrative area, locality, postcode, and geolocation. |

## Platform Data Model

### CPE23PlatformSpecificationType Class

The CPE23PlatformSpecificationType class is defined as a subclass to extend the cybox\_common:PlatformSpecificationType superclass and belongs to the cpe\_2.3 package. As shown in **Figure 3‑2**, the CPE23PlatformSpecificationType class imports and leverages version 2.3 of the Common Platform Enumeration (CPE) [**[CPE]**](#CPE) schema for structured characterization of platforms.

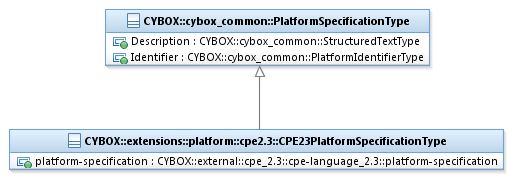


Figure ‑. UML diagram for the CPE23PlatformSpecificationType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **platform-specification** | cpe:platformSpecificationType | 0..1 | The platform-specification property, defined in the CPE 2.3 Applicability Language schema, supports a structured characterization of a platform or combination of platforms. |

# Conformance

Implementations have discretion over which parts (components, properties, extensions, controlled vocabularies, etc.) of CybOX they implement (e.g., Observable/Object).

[1] Conformant implementations must conform to all normative structural specifications of the UML model and to additional normative statements within this document that apply to the portions of CybOX they implement (e.g., implementers of the entire Observable class must conform to all normative structural specifications of the UML model regarding the Observable class, and to additional normative statements contained in the document that describes the Observable class).

[2] Conformant implementations are free to ignore normative structural specifications of the UML model and any additional normative statements within this document that do not apply to the portions of CybOX they implement (e.g., non-implementers of any particular properties of the Observable class are free to ignore all normative structural specifications of the UML model regarding those properties of the Observable class, and any additional normative statements contained in the document that describes the Observable class).

The conformance section of this document is intentionally broad and attempts to reiterate what already exists in this document.

1. 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)

Liron Schiff, Comilion (mobile) Ltd.

Jane Ginn, Cyber Threat Intelligence Network, Inc. (CTIN)

Richard Struse, DHS Office of Cybersecurity and Communications (CS&C)

Ryusuke Masuoka, Fujitsu Limited

Eric Burger, Georgetown University

Jason Keirstead, IBM

Paul Martini, iboss, Inc.

Jerome Athias, Individual

Sanjiv Kalkar, 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

Jonathan Baker, MITRE Corporation

Sean Barnum, MITRE Corporation

Mark Davidson, MITRE Corporation

Ivan Kirillov, 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.

Andrew Storms, New Context Services, Inc.

Igor Baikalov, Securonix

Bernd Grobauer, Siemens AG

John Anderson, Soltra

Trey Darley, Soltra

Paul Dion, Soltra

Brandon Hanes, Soltra

Ali Khan, Soltra

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

1. Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Editor** | **Changes Made** |
| wd01 | 15 December 2015 | Desiree Beck  Trey Darley  Ivan Kirillov  Rich Piazza | Initial transfer to OASIS template |