CybOXTM Version 2.1.1 Part 63: Whois Object

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*. [URI]
* *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*. [URI]
* *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 Session 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*. (this document)
* *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 Whois Object data model, which is one of the Object data models for 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/part1-overview/stix-v1.2.1-csd01-part1-overview.docx

Permanent “Latest version” URI:  
http://docs.oasis-open.org/cti/stix/v1.2.1/stix-v1.2.1-part1-overview.docx

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

Copyright © OASIS Open 2015. 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 2012-2015.  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](#_Toc438560413)

[1.1 CybOXTM Specification Documents 6](#_Toc438560414)

[1.2 Document Conventions 6](#_Toc438560415)

[1.2.1 Fonts 6](#_Toc438560416)

[1.2.2 UML Package References 7](#_Toc438560417)

[1.2.3 UML Diagrams 7](#_Toc438560418)

[1.2.3.1 Class Properties 7](#_Toc438560419)

[1.2.3.2 Diagram Icons and Arrow Types 7](#_Toc438560420)

[1.2.4 Property Table Notation 8](#_Toc438560421)

[1.2.5 Property and Class Descriptions 8](#_Toc438560422)

[1.3 Terminology 9](#_Toc438560423)

[1.4 Normative References 9](#_Toc438560424)

[2 Background Information 10](#_Toc438560425)

[2.1 Cyber Observables 10](#_Toc438560426)

[2.2 Objects 10](#_Toc438560427)

[3 Data Model 11](#_Toc438560428)

[3.1 WhoisObjectType Class 11](#_Toc438560429)

[3.2 WhoisRegistrarInfoType Class 13](#_Toc438560430)

[3.3 WhoisContactsType Class 14](#_Toc438560431)

[3.4 WhoisContactType Data Type 14](#_Toc438560432)

[3.5 WhoisStatusesType Class 15](#_Toc438560433)

[3.6 WhoisStatusType Data Type 16](#_Toc438560434)

[3.7 WhoisNameserversType Class 16](#_Toc438560435)

[3.8 WhoisRegistrantInfoType Class 16](#_Toc438560436)

[3.9 WhoisRegistrantsType Class 16](#_Toc438560437)

[3.10 WhoisStatusTypeEnum Enumeration 17](#_Toc438560438)

[3.11 WhoisDNSSECTypeEnum Enumeration 19](#_Toc438560439)

[3.12 WhoisContactTypeEnum Enumeration 19](#_Toc438560440)

[4 Conformance 20](#_Toc438560441)

[Acknowledgments 21](#_Toc438560442)

[Revision History 22](#_Toc438560443)

# Introduction

[All text is normative unless otherwise labeled]

The Cyber Observable Expression (CybOXTM) 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 Whois Object Version 2.1.1 data model, which is one of eighty-eight CybOX Object data models.

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 Section **1.4**. In Section **2**, we give background information necessary to fully understand the Whois Object data model. We present the Whois Object 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 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 languages, 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. 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 package\_prefix for the Whois data model is WhoisObj. Note that in this specification document, we do not explicitly specify the package prefix for any classes that originate from the Whois Object data model.

### 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, especially in the main top-level component diagrams. 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 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. |

### 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 Whois Object 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>.

# Background Information

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

## Cyber Observables

A cyber observable is a dynamic event or a stateful property that occurs, or may occur, in the operational cyber domain. Examples of stateful properties include the value of a registry key, the MD5 hash of a file, and an IP address. Examples of events include the deletion of a file, the receipt of an HTTP GET request, and the creation of a remote thread.

A cyber observable is different than a cyber indicator. A cyber observable is a statement of fact, capturing what was observed or could be observed in the cyber operational domain. Cyber indicators are cyber observable patterns, such as a registry key value associated with a known bad actor or a spoofed email address used on a particular date.

## Objects

Objects in CybOX are individual data models for characterizing a particular cyber entity, such as a Windows registry key, or an Email Message. Accordingly, each release of the CybOX language includes a particular set of Objects that are part of the release. The data model for each of these Objects is defined by its own specification that describes the context-specific classes and properties that compose the Object.

# Data Model

## WhoisObjectType Class

The WhoisObjectType class is intended to characterize Whois information for a domain. The UML diagram corresponding to the WhoisObjectType class is shown in **Figure 3‑1**.

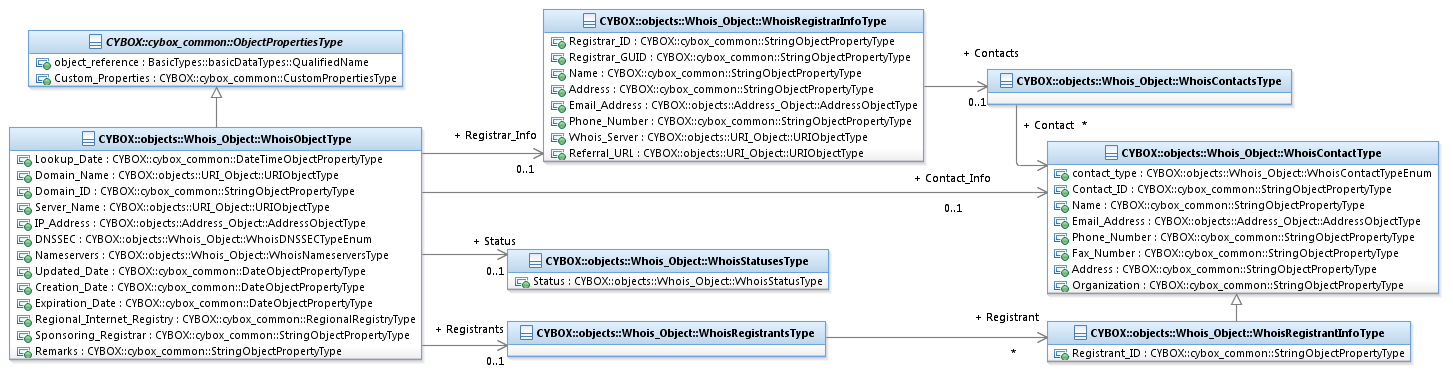


Figure 3‑1. UML diagram of the WhoisObjectType class

The property table of the WhoisObjectType class is given in **Table 3‑1**.

Table 3‑1. Properties of the WhoisObjectType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **Lookup\_Date** | cyboxCommon:  DateTimeObjectPropertyType | 0..1 | The Lookup\_Date property specifies the date and time that the Whois record was queried. |
| **Domain\_Name** | URIObj:URIObjectType | 0..1 | The Domain\_Name property specifies the corresponding domain name for this Whois entry. |
| **Domain\_ID** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Domain\_ID property specifies the domain id for the domain associated with this Whois entry. |
| **Server\_Name** | URIObj:URIObjectType | 0..1 | The Server\_Name property specifies the corresponding server name for this Whois entry. This usually corresponds to a nameserver lookup. |
| **IP\_Address** | AddressObj:  AddressObjectType | 0..1 | The IP\_Address property specifies the corresponding ip address for this Whois entry. The usually corresponds to a nameserver lookup. |
| **DNSSEC** | WhoisDNSSECTypeEnum | 0..1 | The DNSSEC element corresponds to the DNSSEC property associated with a Whois entry. Acceptable values are: "Signed" or "Unsigned". |
| **Nameservers** | WhoisNameserversType | 0..1 | The Nameservers property represents a list of nameserver entries for a Whois entry. |
| **Status** | WhoisStatusesType | 0..1 | The Status property represents a list of statuses for a given Whois entry. |
| **Updated\_Date** | cyboxCommon:  DateObjectPropertyType | 0..1 | The Updated\_Date property specifies the date in which the registered domain information was last updated. |
| **Creation\_Date** | cyboxCommon:  DateObjectPropertyType | 0..1 | The Creation\_Date property specifies the date in which the registered domain was created. |
| **Expiration\_Date** | cyboxCommon:  DateObjectPropertyType | 0..1 | The Expiration\_Date property specifies the date in which the registered domain will expire. |
| **Regional\_Internet\_Registry** | cyboxCommon:  RegionalRegistryType | 0..1 | The Regional\_Internet\_Registry property specifies the name of the Regional Internet Registry (RIR) which allocated the IP address contained in this Whois entry. |
| **Sponsoring\_Registrar** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Sponsoring\_Registrar property holds the name of the sponsoring registrar for the domain. |
| **Registrar\_Info** | WhoisRegistrarInfoType | 0..1 | The Registrar\_Info property represents registrar info that would be returned from a registrar lookup. |
| **Registrants** | WhoisRegistrantsType | 0..1 | The Registrants property represents the registrant information associated with a domain lookup. |
| **Contact\_Info** | WhoisContactType | 0..1 | The Contact\_Info property represents contact info that would be returned from a contact lookup. |
| **Remarks** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Remarks property specifies any remarks associated with this Whois entry. |

## WhoisRegistrarInfoType Class

The property table of the WhoisRegistrarInfoType class is given in **Table 3‑2**.

Table 3‑2. Properties of the WhoisRegistrarInfoType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **Registrar\_ID** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Registrar\_ID corresponds to the Registrar ID property of a Whois entry. |
| **Registrar\_GUID** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Registrar\_GUID corresponds to the Registrar GUID property of a Whois entry. |
| **Name** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Name property holds the name of the registrar organization. |
| **Address** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Address property holds the address (location) of the registrar organization. |
| **Email\_Address** | AddressObj:  AddressObjectType | 0..1 | The Email\_Address property holds the main email address for the registrar. |
| **Phone\_Number** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Phone\_Number property holds the phone number of the registrar organization. |
| **Whois\_Server** | URIObj:URIObjectType | 0..1 | The Whois\_Server property specifies the corresponding whois server for this registrar. |
| **Referral\_URL** | URIObj:URIObjectType | 0..1 | The Referral\_URL property specifies the corresponding referral URL for registrar. |
| **Contacts** | WhoisContactsType | 0..1 | The Contacts property specifies a list of registrar contacts. |

## WhoisContactsType Class

The WhoisContactsType class represents a list of contacts (usually registrar or registrant) found in a Whois entry.

The property table of the WhoisContactsType class is given in **Table 3‑3**.

Table 3‑3. Properties of the WhoisContactsType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **Contact** | WhoisContactType | 0..\* | The Contact property specifies a contact found in a Whois entry. |

## WhoisContactType Data Type

The property table of the WhoisContactType class is given in **Table 3‑4**.

Table 3‑4. Properties of the WhoisContactType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **contact\_type** | WhoisContactTypeEnum | 0..1 | The contact\_type property specifies what type of contact this is. |
| **Contact\_ID** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Contact\_ID property corresponds to an ID for the contact. This can be presented as Contact ID, Billing ID, Admin ID, Tech ID, etc. |
| **Name** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Name property specifies the name of the contact. |
| **Email\_Address** | AddressObj:  AddressObjectType | 0..1 | The Email\_Address property specifies the email address of the contact. |
| **Phone\_Number** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Phone\_Number property specifies the phone number of the contact. |
| **Fax\_Number** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Fax\_Number property specifies the fax number of the contact. |
| **Address** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Address property specifies the address of the contact. |
| **Organization** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Organization property specifies the name of the organization this contact works for or is associated with. |

## WhoisStatusesType Class

The WhoisStatusesType class defines a list of WhoisStatusType objects.

The property table of the WhoisStatusesType class is given in **Table 3‑5**.

Table 3‑5. Properties of the WhoisStatusesType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **Status** | WhoisObj:WhoisStatusType | 0..\* | The Status property |

## WhoisStatusType Data Type

The WhoisStatusType data type specifies a status for a domain as listed in its Whois entry. Its core value SHOULD be a literal found in the WhoisStatusTypeEnum enumeration. Its base type is the BaseObjectPropertyType data type, in order to permit complex (i.e. regular-expression based) specifications.

## WhoisNameserversType Class

The WhoisNameserversType class defines a list of nameservers associated with a Whois entry.

The property table of the WhoisNameserversType class is given in **Table 3‑6**.

Table 3‑6. Properties of the WhoisNameserversType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **Nameserver** | URIObj:URIObjectType | 0..\* | The Nameserver property specifies a nameserver of the domain for this whois entry. |

## WhoisRegistrantInfoType Class

The property table of the WhoisRegistrantInfoType class is given in **Table 3‑7**.

Table 3‑7. Properties of the WhoisRegistrantInfoType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **Registrant\_ID** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Registrant\_ID property specifies the registrant id for a given registrant. |

## WhoisRegistrantsType Class

The WhoisRegistrantsType class represents a list of registrant information for a given Whois entry.

The property table of the WhoisRegistrantsType class is given in **Table 3‑8**.

Table 3‑8. Properties of the WhoisRegistrantsType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **Registrant** | WhoisRegistrantInfoType | 0..\* | The Registrant property |

## WhoisStatusTypeEnum Enumeration

The literals of the WhoisStatusTypeEnum enumeration are given in **Table 3‑9**.

Table 3‑9. Literals of the WhoisStatusTypeEnum enumeration

|  |  |
| --- | --- |
| **Enumeration Literal** | **Description** |
| **ADD\_PERIOD** | The 5-day Add Grace Period after the initial registration of a domain. If the domain is deleted by the registrar during this period, the registry provides a credit to the registrar for the cost of the registration. |
| **RENEW\_PERIOD** | The 5-day period after a domain registration period is explicitly extended (renewed) by the registrar. If the domain is deleted by the registrar during this period, the registry provides a credit to the registrar for the cost of the renewal. |
| **AUTO\_RENEW\_PERIOD** | The 45-day period after a domain registration period expires and is extended (renewed) automatically by the registry. If the domain is deleted by the registrar during this period, the registry provides a credit to the registrar for the cost of the renewal. |
| **TRANSFER\_PERIOD** | The 5-day period after the successful transfer of domain name registration sponsorship from one registrar to another registrar. If the domain is deleted by the new sponsoring registrar during this period, the registry provides a credit to the registrar for the cost of the transfer. |
| **PENDING\_DELETE\_RESTORABLE** | The 30-day period after a registrar has submitted a delete command to delete a domain from the registry. All Internet services associated with the domain are disabled. During this period, a registrar can submit a request to Restore the domain. |
| **PENDING\_DELETE\_**  **SCHEDULED\_FOR\_RELEASE** | The 5-day period following the PENDING DELETE RESTORABLE period. During this period, all Internet services associated with the domain will remain disabled and domain cannot be Restored. |
| **PENDING\_RESTORE** | The registrar has submitted a Restore request for a domain that was previously in the status of PENDING DELETE RESTORABLE and the registry is awaiting a Restore Report from the registrar. |
| **OK** | This is the normal status for a domain that has no pending operations or prohibitions. |
| **INACTIVE** | The domain has no associated nameservers. A minimum of 2 nameservers must be associated with the domain before it can be published to the zone. |
| **CLIENT\_TRANSFER\_**  **PROHIBITED** | Registrar does not allow the transfer of a domain. |
| **CLIENT\_RENEW\_PROHIBITED** | Registrar does not allow the renewal of a domain. |
| **CLIENT\_DELETE\_PROHIBITED** | Registrar does not allow the deletion of a domain. |
| **CLIENT\_UPDATE\_PROHIBITED** | Registrar does not allow the update or modification of a domain. |
| **CLIENT\_HOLD** | Registrar will not allow the domain to be published to the zone. |
| **TRANSFER\_PROHIBITED** | Registry does not allow the transfer of a domain. |
| **RENEW\_PROHIBITED** | Registry does not allow the renewal of a domain. |
| **DELETE\_PROHIBITED** | Registry does not allow the deletion of a domain. |
| **UPDATE\_PROHIBITED** | Registry does not allow all the update or modification of a domain. |
| **HOLD** | Registry will not allow the domain to be published to the zone. |

## WhoisDNSSECTypeEnum Enumeration

The literals of the WhoisDNSSECTypeEnum enumeration are given in **Table 3‑10**.

Table 3‑10. Literals of the WhoisDNSSECTypeEnum enumeration

|  |  |
| --- | --- |
| **Enumeration Literal** | **Description** |
| **Signed** | The Signed value signifies that the domain name associated with the Whois entry is digitally signed. |
| **Unsigned** | The Unsigned value signifies that the domain name associated with the Whois entry is not digitally signed. |

## WhoisContactTypeEnum Enumeration

The literals of the WhoisContactTypeEnum enumeration are given in **Table 3‑11**.

Table 3‑11. Literals of the WhoisContactTypeEnum enumeration

|  |  |
| --- | --- |
| **Enumeration Literal** | **Description** |
| **ADMIN** | The contact is an administrator. |
| **BILLING** | The contact is for billing. |
| **TECHNICAL** | The contact is for technical assistance. |

# 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 or 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 or 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 or 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 or 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.

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.

Revision History

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