CybOX™ Version 2.1.1. Part 05: Vocabularies

Committee Specification Draft 01 / Public Review Draft 01

20 June 2016

Specification URIs

This version:

http://docs.oasis-open.org/cti/cybox/v2.1.1/csprd01/part05-vocabularies/cybox-v2.1.1-csprd01-part05-vocabularies.docx (Authoritative)
http://docs.oasis-open.org/cti/cybox/v2.1.1/csprd01/part05-vocabularies/cybox-v2.1.1-csprd01-part05-vocabularies.html
http://docs.oasis-open.org/cti/cybox/v2.1.1/csprd01/part05-vocabularies/cybox-v2.1.1-csprd01-part05-vocabularies.pdf

Previous version:

N/A

Latest version:

http://docs.oasis-open.org/cti/cybox/v2.1.1/part05-vocabularies/cybox-v2.1.1-part05-vocabularies.docx (Authoritative)
http://docs.oasis-open.org/cti/cybox/v2.1.1/part05-vocabularies/cybox-v2.1.1-part05-vocabularies.html
http://docs.oasis-open.org/cti/cybox/v2.1.1/part05-vocabularies/cybox-v2.1.1-part05-vocabularies.pdf

Technical Committee:

OASIS Cyber Threat Intelligence (CTI) TC

Chair:

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

Editors:

Desiree Beck (dbeck@mitre.org), MITRE Corporation
Trey Darley (trey@kingfisherops.com), Individual member
Ivan Kirillov (ikirillov@mitre.org), MITRE Corporation
Rich Piazza (rpiazza@mitre.org), MITRE Corporation

Additional artifacts:

This prose specification is one component of a Work Product whose components are listed in http://docs.oasis-open.org/cti/cybox/v2.1.1/csprd01/cybox-v2.1.1-csprd01-additional-artifacts.html.

Related work:

This specification is related to:

- *STIX™ Version 1.2.1*. Edited by Sean Barnum, Desiree Beck, Aharon Chernin, and Rich Piazza. 05 May 2016. OASIS Committee Specification 01. http://docs.oasis-open.org/cti/stix/v1.2.1/cs01/part1-overview/stix-v1.2.1-cs01-part1-overview.html.

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 Vocabularies data model, which includes definitions for default constrained enumerations of values for specific properties in other CybOX data models.

Status:

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

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

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the TC's web page (https://www.oasis-open.org/committees/cti/ipr.php).

Citation format:

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

[CybOX-v2.1.1-vocabularies]

CybOX™ Version 2.1.1. Part 05: Vocabularies. Edited by Desiree Beck, Trey Darley, Ivan Kirillov, and Rich Piazza. 20 June 2016. OASIS Committee Specification Draft 01 / Public Review Draft 01. http://docs.oasis-open.org/cti/cybox/v2.1.1/csprd01/part05-vocabularies/cybox-v2.1.1-csprd01-part05-vocabularies.html. Latest version: http://docs.oasis-open.org/cti/cybox/v2.1.1/part05-vocabularies/cybox-v2.1.1-part05-vocabularies.html.

Notices

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

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

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

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

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

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

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

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

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

Table of Contents

1 Introduction 6
1.1 CybOX™ Specification Documents 6
1.2 Document Conventions 6
1.2.1 Fonts 6
1.2.2 UML Package References 7
1.2.3 UML Diagrams 7
1.2.4 Enumeration Table Notation 8
1.3 Terminology 8
1.4 Normative References 8
2 Background Information 9
2.1.1 VocabularyStringType Data Type 11
2.1.2 UnenforcedVocabularyStringType Data Type 11
2.1.3 ControlledVocabularyStringType Data Type 11
3 CybOX Default Vocabularies Data Models 12
3.1 ActionTypeVocab-1.0 Enumeration 12
3.2 ActionNameVocab-1.1 Enumeration 18
3.3 ActionNameVocab-1.0 Enumeration 27
3.4 ActionArgumentNameVocab-1.0 Enumeration 36
3.5 ActionObjectAssociationTypeVocab-1.0 Enumeration 39
3.6 ActionRelationshipTypeVocab-1.0 Enumeration 39
3.7 EventTypeVocab-1.0.1 Enumeration 40
3.8 EventTypeVocab-1.0 Enumeration 42
3.9 ObjectRelationshipVocab-1.1 Enumeration 44
3.10 ObjectRelationshipVocab-1.0 Enumeration 51
3.11 ObjectStateVocab-1.0 Enumeration 57
3.12 CharacterEncodingVocab-1.0 Enumeration 58
3.13 InformationSourceTypeVocab-1.0 Enumeration 59
3.14 HashNameVocab-1.0 Enumeration 60
3.15 ToolTypeVocab-1.1 Enumeration 60
3.16 ToolTypeVocab-1.0 Enumeration 62
4 Conformance 64
Appendix A. Acknowledgments 65
Appendix B. Revision History 69

1 Introduction

[All text is normative unless otherwise labeled.]

The Cyber Observable Expression (CybOX™) 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 Vocabularies Version 2.1.1 data model, which is one of ninety-three data models for 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 Section **1.4**. In Section **2**, we give background information necessary to fully understand the Vocabularies data model. We present the Vocabularies data model specification details in Section **3**, and conformance information in Section **4**.

1.1 CybOX™ 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* 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* also summarizes the relationship of CybOX to other languages, and outlines general CybOX data model conventions.

1.2 Document Conventions

The following conventions are used in this document.

1.2.1 Fonts

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

- Capitalization is used for CybOX high-level concepts, which are defined in *CybOX™ Version 2.1.1 Part 1: Overview*.

  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 (with single 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*

1.2.2 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.

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

1.2.3 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.

1.2.3.1 Diagram Icons and Arrow Types

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

Table 1-1. UML diagram icons

Icon	Description
	This diagram icon indicates a class. If the name is in italics, it is an abstract class.
	This diagram icon indicates an enumeration.

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

1.2.4 Enumeration Table Notation

Throughout Section **3**, tables are used to describe the list of defined values for each default vocabulary. Each property table consists of a column of literal names, and a description column that describes the literal name, if needed.

1.3 Terminology

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

1.4 Normative References

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

2 Background Information

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

There are three vocabulary-related UML data types defined in the Common data model, and together they provide a content creator with four choices for defining content, listed below in order of formality:

- Leverage a default vocabulary using the `ControlledVocabularyStringType` data type. CybOX v1.2.1 defines a collection of default vocabularies and associated enumerations that are based on input from the CybOX; however, not all vocabulary properties have an assigned default vocabulary.

- Formally define a custom vocabulary using the `ControlledVocabularyStringType` data type. To achieve value enforcement, a custom vocabulary MUST be formally added to the CybOX Vocabulary data model. Because this is an extension of the CybOX Vocabulary data model, producers and consumers MUST be aware of the addition to the data model for successful sharing of CybOX documents.

- Reference an externally-defined, custom vocabulary using the `UnenforcedVocabularyStringType` data type to constrain the set of values. Externally-defined vocabularies are publically defined, but have not been included as formally specified vocabularies within the CybOX Vocabulary data model using the `ControlledVocabularyStringType` data type. In this case, it is sufficient to specify the name of the vocabulary and a URL that defines that vocabulary.

- Choose an arbitrary and unconstrained value using the `VocabularyStringType` data type.

While not required by the general CybOX language, default vocabularies should be used whenever possible to ensure the greatest level of compatibility between CybOX users. If an appropriate default vocabulary is not available, a formally defined custom vocabulary can be specified and leveraged. In addition to compatibility advantages, using formally defined vocabularies (whether default vocabularies or otherwise defined) enables enforced use of valid enumeration values.

If a formally defined vocabulary is not sufficient for a content producer's purposes, the CybOX Vocabulary data model allows the two alternatives listed above: externally defined custom vocabularies and arbitrary string values, which dispense with enumerated vocabularies altogether. If a custom vocabulary is not formally added to the Vocabulary data model, then no enforcement policy of appropriate values is specified.

The base data type of the `VocabularyStringType` data type is a `BasicString` from the `BasicTypes` package. Additionally, `VocabularyStringType` is also a sub data type of `cyboxCommon:PatternFieldGroup`, in order to permit complex (i.e. regular-expression based) specifications.

The UML diagram shown in **Figure 2-1** illustrates the relationships between the three vocabulary data types defined in the CybOX Common data model. As illustrated, all controlled vocabularies formally defined within the CybOX Vocabulary data model are defined using an enumeration derived from the `ControlledVocabularyStringType` data type.

As shown, the `HashNameVocab-1.0` enumeration (used as a defined controlled vocabulary exemplar) is defined as a specialization of the `ControlledVocabularyStringType` data type, and therefore it is also a specialization of the `VocabularyStringType` data type.

Further details of each vocabulary class are provided in Subsections **2.1.1** through **2.1.3**.

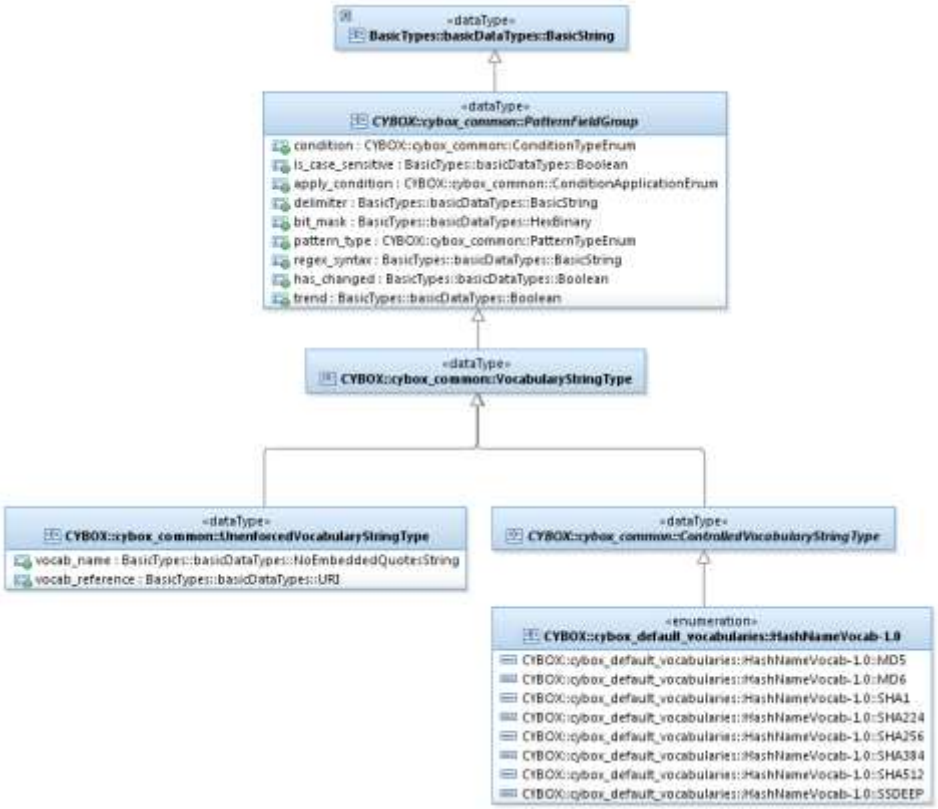

Figure 2-1. UML diagram of the CybOX Vocabulary data model

2.1.1 VocabularyStringType Data Type

The `VocabularyStringType` data type is the basic data type of all vocabularies. Therefore, all properties in the collection of CybOX data models that makes use of the Vocabulary data model must be defined to use the `VocabularyStringType` data type. Because this data type is a specialization of the `basicDataTypes:BasicString` data type, it can be used to support the arbitrary string option for vocabularies.

2.1.2 UnenforcedVocabularyStringType Data Type

The `UnenforcedVocabularyStringType` data type specifies custom vocabulary values via an enumeration defined outside of the CybOX Vocabulary data model. It extends the `VocabularyStringType` data type. Note that the CybOX vocabulary data model does not define any enforcement policy for this data type.

The property table of the `UnenforcedVocabularyStringType` data type is given in **Table 2-1**.

Table 2-1. Properties of the `UnenforcedVocabularyStringType` data type

Name	Type	Multiplicity	Description
vocab_name	`basicDataTypes:NoEmbeddedQuoteString`	0..1	The `vocab_name` property specifies the name of the externally defined vocabulary.
vocab_reference	`basicDataTypes:URI`	0..1	The `vocab_reference` property specifies the location of the externally defined vocabulary using a Uniform Resource Identifier (URI).

2.1.3 ControlledVocabularyStringType Data Type

The `ControlledVocabularyStringType` data type specifies a formally defined vocabulary. It is an abstract data type so it MUST be extended via an enumeration from the CybOX Vocabulary data model. Any custom vocabulary must be defined via an enumeration added to the CybOX Vocabulary data model, if appropriate enumeration values are to be enforced.

The `ControlledVocabularyStringType` class has no properties of its own, so there is no associated property table.

3 CybOX Default Vocabularies Data Models

The CybOX Vocabularies data model is defined as one UML package, but can be thought of as a collection of separate data models, each containing one UML enumeration. Each vocabulary will be specified using a separate version number, which is appended to the enumeration name. This facilitates adding literals to the enumeration without the need to update the version number of any of the other CybOX data models or the version number of the full CybOX specification.

3.1 ActionTypeVocab-1.0 Enumeration

The `ActionTypeVocab` enumeration is the default CybOX vocabulary for Action classes, captured via the `ActionType` class (`Type` property) in CybOX Core. The associated enumeration literals are provided in the table below.

Enumeration Literal	Description
Accept	Specifies the atomic action of accepting an object or value.
Access	Specifies the atomic action of accessing an object.
Add	Specifies the atomic action of adding an object.
Alert	Specifies the atomic action of issuing an alert.
Allocate	Specifies the atomic action of allocating an object.
Archive	Specifies the atomic action of archiving an object or data.
Assign	Specifies the atomic action of assigning a value to an object.
Audit	Specifies the atomic action of auditing an object or data.
Backup	Specifies the atomic action of backing up an object or data.
Bind	Specifies the atomic action of binding two objects.
Block	Specifies the atomic action of blocking access to an object or resource.
Call	Specifies the atomic action of calling an object or resource.
Change	Specifies the atomic action of changing an object.
Check	Specifies the atomic action of checking an object.

Clean	Specifies the atomic action of cleaning an object, such as a file system.
Click	Specifies the atomic action of clicking an object, as with a mouse.
Close	Specifies the atomic action of closing an object, such as a window handle.
Compare	Specifies the atomic action of comparing two objects.
Compress	Specifies the atomic action of compressing an object.
Configure	Specifies the atomic action of configuring a resource.
Connect	Specifies the atomic action of connecting to an object, such as a service or resource.
Control	Specifies the atomic action of controlling an object or data.
Copy/Duplicate	Specifies the atomic action of copying or duplicating an object or data EXCEPT in cases where the object is considered a thread or process as a whole.
Create	Specifies the atomic action of creating an object or data.
Decode	Specifies the atomic action of decoding an object or data.
Decompress	Specifies the atomic action of decompressing an object, such as an archive.
Decrypt	Specifies the atomic action of decrypting an object.
Deny	Specifies the atomic action of denying access to an object or resource.
Depress	Specifies the atomic action of depressing an object that has been pressed, such a button.
Detect	Specifies the atomic action of detecting an object.
Disconnect	Specifies the atomic action of disconnecting from a service or resource.
Download	Specifies the atomic action of downloading an object or data.
Draw	Specifies the atomic action of drawing an object.
Drop	Specifies the atomic action of dropping an object, such as a connection.

Encode	Specifies the atomic action of encoding an object or data.
Encrypt	Specifies the atomic action of encrypting an object or data.
Enumerate	Specifies the atomic action of enumerating a list of objects.
Execute	Specifies the atomic action of executing an object, such as an executable file.
Extract	Specifies the atomic action of extracting an object.
Filter	Specifies the atomic action of filtering an object or data.
Find	Specifies the atomic action of finding an object or data.
Flush	Specifies the atomic action of flushing an object or data, such as a cache.
Fork	Specifies the atomic action of forking, as with a process. Because this is usually associated with processes and threads and does not generalize to objects, it is DIFFERENT from Copy/Duplicate.
Free	Specifies the atomic action of freeing an object.
Get	Specifies the atomic action of getting a value from an object.
Hook	Specifies the atomic action of hooking an object to another object.
Hide	Specifies the atomic action of hiding an object.
Impersonate	Specifies the atomic action of impersonation, in which an object performs actions that assume the character or appearance of another object.
Initialize	Specifies the atomic action of initializing an object.
Inject	Specifies the atomic action of injecting an object.
Install	Specifies the atomic action of installing an object, such as an application, program, patch, or other resource.
Interleave	Specifies the atomic action of interleaving an object, that is, the action of arranging data in a non-contiguous way to increase performance.
Join	Specifies the atomic action of joining one object to another object.

Kill	Specifies the atomic action of killing an object, as with a thread or program.
Listen	Specifies the atomic action of listening to an object, such as to a port on a network connection.
Load	Specifies the atomic action of loading an object.
Lock	Specifies the atomic action of locking an object.
Login/Logon	Specifies the atomic action of logging into an object, such as into a system or application.
Logout/Logoff	Specifies the atomic action of logging out of an object, such as a system or application.
Map	Specifies the atomic action of mapping an object to another object or data.
Merge	Specifies the atomic action of merging one object to another object.
Modify	Specifies the atomic action of modifying an object.
Monitor	Specifies the atomic action of monitoring the state of an object.
Move	Specifies the atomic action of moving an object.
Open	Specifies the atomic action of opening an object.
Pack	Specifies the atomic action of packing an object.
Pause	Specifies the atomic action of pausing an object, such as a thread or process.
Press	Specifies the atomic action of pressing an object, such as a button.
Protect	Specifies the atomic action of protecting an object.
Quarantine	Specifies the atomic action of placing an object in quarantine, that is, to store the object in an isolated area away from other objects so it can be safely operated on.
Query	Specifies the atomic action of querying an object.
Queue	Specifies the atomic action of queueing an object.

Raise	Specifies the atomic action of raising an object.
Read	Specifies the atomic action of reading an object.
Receive	Specifies the atomic action of receiving an object.
Release	Specifies the atomic action of releasing an object.
Rename	Specifies the atomic action of renaming an object.
Remove/Delete	Specifies the atomic action of removing or deleting an object.
Replicate	Specifies the atomic action of replicating an object.
Restore	Specifies the atomic action of restoring an object.
Resume	Specifies the atomic action of resuming an object, as with a process or thread.
Revert	Specifies the atomic action of reverting an object.
Run	Specifies the atomic action of running an object, such as an application.
Save	Specifies the atomic action of saving an object.
Scan	Specifies the atomic action of scanning for an object or data.
Schedule	Specifies the atomic action of scheduling an object, such as an event.
Search	Specifies the atomic action of searching for an object.
Send	Specifies the atomic action of sending an object.
Set	Specifies the atomic action of setting an object to a value.
Shutdown	Specifies the atomic action of shutting down an object.
Sleep	Specifies the atomic action of putting to sleep an object.
Snapshot	Specifies the atomic action taking a snapshot of an object.
Start	Specifies the atomic action of starting an object, such as a thread or process.

Stop	Specifies the atomic action of stopping an object, such as a thread or process.
Suspend	Specifies the atomic action of suspending an object, such an account or privileges for an account.
Synchronize	Specifies the atomic action of synchronizing an object.
Throw	Specifies the atomic action of throwing an object, such as an exception in a programming language.
Transmit	Specifies the atomic action of transmitting an object.
Unblock	Specifies the atomic action of unblocking an object.
Unhide	Specifies the atomic action of unhiding an object.
Unhook	Specifies the atomic action of unhooking an object from another object, that is, to detach.
Uninstall	Specifies the atomic action of uninstalling an object.
Unload	Specifies the atomic action of unloading an object.
Unlock	Specifies the atomic action of unlocking an object.
Unmap	Specifies the atomic action of unmapping an object from another object or data.
Unpack	Specifies the atomic action of unpacking an object, such as an archive.
Update	Specifies the atomic action of updating an object.
Upgrade	Specifies the atomic action of upgrading an object.
Upload	Specifies the atomic action of uploading an object.
Wipe/Destroy/Purge	Specifies the atomic action of wiping, destroying, or purging an object.
Write	Specifies the atomic action of writing an object.

3.2 ActionNameVocab-1.1 Enumeration

The `ActionNameVocab` enumeration is the default CybOX vocabulary for Action names, captured via the `ActionType` class (`Name` property) in CybOX Core. The associated enumeration literals are provided in the table below.

Enumeration Literal	Description
Accept Socket Connection	Specifies the defined action of accepting a socket connection.
Add Connection to Network Share	Specifies the defined action of adding a connection to an existing network share.
Add Network Share	Specifies the defined action of adding a new network share.
Add System Call Hook	Specifies the defined action of adding a new system call hook.
Add User	Specifies the defined action of adding a new user.
Add Windows Hook	Specifies the defined action of adding a new Windows hook.
Add Scheduled Task	Specifies the defined action of adding a scheduled task.
Allocate Virtual Memory in Process	Specifies the defined action of allocating virtual memory in a process.
Bind Address to Socket	Specifies the defined action of binding an address to a socket.
Change Service Configuration	Specifies the defined action of changing the service configuration.
Check for Remote Debugger	Specifies the defined action of checking for a remote debugger.
Close Port	Specifies the defined action of closing a port.
Close Registry Key	Specifies the defined action of closing a registry key.
Close Socket	Specifies the defined action of closing a socket.
Configure Service	Specifies the defined action of configuring a service.
Connect to IP	Specifies the defined action of connecting to an IP address.

Connect to Named Pipe	Specifies the defined action of connecting to a named pipe.
Connect to Network Share	Specifies the defined action of connecting to a network share.
Connect to Socket	Specifies the defined action of connecting to a socket.
Connect to URL	Specifies the defined action of connecting to a URL.
Control Driver	Specifies the defined action of controlling a driver.
Control Service	Specifies the defined action of controlling a service.
Copy File	Specifies the defined action of copying a file.
Create Dialog Box	Specifies the defined action of creating a dialog box.
Create Directory	Specifies the defined action of creating a new directory.
Create Event	Specifies the defined action of creating an event.
Create File	Specifies the defined action of creating a file.
Create File Alternate Data Stream	Specifies the defined action of creating an alternate data stream in a file.
Create File Mapping	Specifies the defined action of creating a new file mapping.
Create File Symbolic Link	Specifies the defined action of creating a file symbolic link.
Create Hidden File	Specifies the defined action of creating a hidden file.
Create Mailslot	Specifies the defined action of creating a mailslot.
Create Module	Specifies the defined action of creating a module.
Create Mutex	Specifies the defined action of creating a mutex.
Create Named Pipe	Specifies the defined action of creating a named pipe.
Create Process	Specifies the defined action of creating a process.
Create Process as User	Specifies the defined action of creating a process as user.

Create Registry Key	Specifies the defined action of creating a registry key.
Create Registry Key Value	Specifies the defined action of creating a registry key value.
Create Remote Thread in Process	Specifies the defined action of creating a remote thread in a process.
Create Service	Specifies the defined action of creating a service.
Create Socket	Specifies the defined action of creating a socket.
Create Symbolic Link	Specifies the defined action of creating a symbolic link.
Create Thread	Specifies the defined action of creating a thread.
Create Window	Specifies the defined action of creating a window.
Delete Directory	Specifies the defined action of deleting a directory.
Delete File	Specifies the defined action of deleting a file.
Delete Named Pipe	Specifies the defined action of deleting a named pipe.
Delete Network Share	Specifies the defined action of deleting a network share.
Delete Registry Key	Specifies the defined action of deleting a registry key.
Delete Registry Key Value	Specifies the defined action of deleting a registry key value.
Delete Service	Specifies the defined action of deleting a service.
Delete User	Specifies the defined action of deleting a user.
Disconnect from Named Pipe	Specifies the defined action of disconnecting from a named pipe.
Disconnect from Network Share	Specifies the defined action of disconnecting from a network share.
Disconnect from Socket	Specifies the defined action of disconnecting from a socket.
Download File	Specifies the defined action of downloading a file.

Enumerate DLLs	Specifies the defined action of enumerating DLLs.
Enumerate Network Shares	Specifies the defined action of enumerating network shares.
Enumerate Protocols	Specifies the defined action of enumerating protocols.
Enumerate Registry Key Subkeys	Specifies the defined action of enumerating registry key subkeys.
Enumerate Registry Key Values	Specifies the defined action of enumerating registry key values.
Enumerate Threads in Process	Specifies the defined action of enumerating threads in a process.
Enumerate Processes	Specifies the defined action of enumerating processes.
Enumerate Services	Specifies the defined action of enumerating services.
Enumerate System Handles	Specifies the defined action of enumerating system handles.
Enumerate Threads	Specifies the defined action of enumerating threads.
Enumerate Users	Specifies the defined action of enumerating users.
Enumerate Windows	Specifies the defined action of enumerating windows.
Find File	Specifies the defined action of finding a file.
Find Window	Specifies the defined action of finding a window.
Flush Process Instruction Cache	Specifies the defined action of flushing the process instruction cache.
Free Library	Specifies the defined action of freeing a library.
Free Process Virtual Memory	Specifies the defined action of freeing virtual memory from a process.
Get Disk Free Space	Specifies the defined action of getting the amount of free space available on a disk.
Get Disk Type	Specifies the defined action of getting the disk type.

Get Elapsed System Up Time	Specifies the defined action of getting the elapsed system up-time.
Get File Attributes	Specifies the defined action of getting file attributes.
Get Function Address	Specifies the defined action of getting the function address.
Get System Global Flags	Specifies the defined action of getting system global flags.
Get Host By Address	Specifies the defined action of getting host by address.
Get Host By Name	Specifies the defined action of getting host by name.
Get Host Name	Specifies the defined action of getting the host name.
Get Library File Name	Specifies the defined action of getting the library file name.
Get Library Handle	Specifies the defined action of getting the library handle.
Get NetBIOS Name	Specifies the defined action of getting the NetBIOS name.
Get Process Current Directory	Specifies the defined action of getting the process's current directory.
Get Process Environment Variable	Specifies the defined action of getting the process environment variable.
Get Process Startup Information	Specifies the defined action of getting the process startup information.
Get Processes Snapshot	Specifies the defined action of getting the processes snapshot.
Get Registry Key Attributes	Specifies the defined action of getting the attributes of a registry key.
Get Service Status	Specifies the defined action of getting the service status.
Get System Global Flags	Specifies the defined action of getting the system global flags.
Get System Local Time	Specifies the defined action of getting the local time on a system.
Get System Host Name	Specifies the defined action of getting the system host name.

Get System NetBIOS Name	Specifies the defined action of getting the NetBIOS name of a system.
Get System Network Parameters	Specifies the defined action of getting the system network parameters.
Get System Time	Specifies the defined action of getting the system time.
Get Thread Context	Specifies the defined action of getting the thread context.
Get Thread Username	Specifies the defined action of getting the thread username.
Get User Attributes	Specifies the defined action of getting the attributes of a user.
Get Username	Specifies the defined action of getting a username.
Get Windows Directory	Specifies the defined action of getting a Windows directory.
Get Windows System Directory	Specifies the defined action of getting a Windows directory.
Get Windows Temporary Files Directory	Specifies the defined action of getting the Windows temporary files directory.
Hide Window	Specifies the defined action of hiding a window.
Impersonate Process	Specifies the defined action of impersonating a process.
Impersonate Thread	Specifies the defined action of impersonating a thread.
Inject Memory Page	Specifies the defined action of injecting a memory page into a process.
Kill Process	Specifies the defined action of killing a process.
Kill Thread	Specifies the defined action of killing a thread.
Kill Window	Specifies the defined action of killing a window.
Listen on Port	Specifies the defined action of listening on a specific port.
Listen on Socket	Specifies the defined action of listening on a socket.
Load and Call Driver	Specifies the defined action of loading and calling a driver.

Load Driver	Specifies the defined action of loading a driver.
Load Library	Specifies the defined action of loading a library.
Load Module	Specifies the defined action of loading a module.
Lock File	Specifies the defined action of locking a file.
Logon as User	Specifies the defined action of logging on as a user.
Map File	Specifies the defined action of mapping a file.
Map Library	Specifies the defined action of mapping a library.
Map View of File	Specifies the defined action of mapping a view of a file.
Modify File	Specifies the defined action of modifying a file.
Modify Named Pipe	Specifies the defined action of modifying a named pipe.
Modify Process	Specifies the defined action of modifying a process.
Modify Service	Specifies the defined action of modifying a service.
Modify Registry Key	Specifies the defined action of modifying a registry key.
Modify Registry Key Value	Specifies the defined action of modifying a registry key value.
Monitor Registry Key	Specifies the defined action of monitoring a registry key.
Move File	Specifies the defined action of moving a file.
Open File	Specifies the defined action of opening a file.
Open File Mapping	Specifies the defined action of opening a file mapping.
Open Mutex	Specifies the defined action of opening a mutex.
Open Port	Specifies the defined action of opening a port.
Open Process	Specifies the defined action of opening a process.

Open Registry Key	Specifies the defined action of opening a registry key.
Open Service	Specifies the defined action of opening a service.
Open Service Control Manager	Specifies the defined action of opening a service control manager.
Protect Virtual Memory	Specifies the defined action of protecting virtual memory.
Query Disk Attributes	Specifies the defined action of querying disk attributes.
Query DNS	Specifies the defined action of querying DNS.
Query Process Virtual Memory	Specifies the defined action of querying process virtual memory.
Queue APC in Thread	Specifies the defined action of querying the Asynchronous Procedure Call (APC) in the context of a thread.
Read File	Specifies the defined action of reading a file.
Read From Named Pipe	Specifies the defined action of reading from a named pipe.
Read From Process Memory	Specifies the defined action of reading from process memory.
Read Registry Key Value	Specifies the defined action of reading a registry key value.
Receive Data on Socket	Specifies the defined action of receiving data on a socket.
Receive Email Message	Specifies the defined action of receiving an email message.
Release Mutex	Specifies the defined action of releasing a mutex.
Rename File	Specifies the defined action of renaming a file.
Revert Thread to Self	Specifies the defined action of reverting a thread to its self.
Send Control Code to File	Specifies the defined action of sending a control code to a file.
Send Control Code to Pipe	Specifies the defined action of sending a control code to a pipe.
Send Control Code to Service	Specifies the defined action of sending control code to a

	service.
Send Data on Socket	Specifies the defined action of sending data on a socket.
Send Data to Address on Socket	Specifies the defined action of sending data to the address on a socket.
Send DNS Query	Specifies the defined action of sending a DNS query.
Send Email Message	Specifies the defined action of sending an email message.
Send ICMP Request	Specifies the defined action of sending an ICMP request.
Send Reverse DNS Query	Specifies the defined action of sending a reverse DNS query.
Set File Attributes	Specifies the defined action of setting file attributes.
Set NetBIOS Name	Specifies the defined action of setting the NetBIOS name.
Set Process Current Directory	Specifies the defined action of setting the process current directory.
Set Process Environment Variable	Specifies the defined action of setting the process environment variable.
Set System Global Flags	Specifies the defined action of setting system global flags.
Set System Host Name	Specifies the defined action of setting the system host name.
Set System Time	Specifies the defined action of setting the system time.
Set Thread Context	Specifies the defined action of setting the thread context.
Show Window	Specifies the defined action of showing a window.
Shutdown System	Specifies the defined action of shutting down a system.
Sleep Process	Specifies the defined action of sleeping a process.
Sleep System	Specifies the defined action of sleeping a system.
Start Service	Specifies the defined action of starting a service.

Unload Driver	Specifies the defined action of unloading a driver.
Unlock File	Specifies the defined action of unlocking a file.
Unmap File	Specifies the defined action of unmapping a file.
Unload Module	Specifies the defined action of unloading a module.
Upload File	Specifies the defined action of uploading a file.
Write to File	Specifies the defined action of writing to a file.
Write to Process Virtual Memory	Specifies the defined action of writing to process virtual memory.

3.3 ActionNameVocab-1.0 Enumeration

The `ActionNameVocab` enumeration is the default CybOX vocabulary for Action names, captured via the `ActionType` class (`Name` property) in CybOX Core. The associated enumeration literals are provided in the table below. NOTE: As of CybOX Version 2.1, `ActionNameVocab-1.0` is deprecated. Please use version 1.1 instead (see Section **3.2**).

Enumeration Literal	Description
Accept Socket Connection	Specifies the defined action of accepting a socket connection.
Add Connection to Network Share	Specifies the defined action of adding a connection to an existing network share.
Add Network Share	Specifies the defined action of adding a new network share.
Add System Call Hook	Specifies the defined action of adding a new system call hook.
Add User	Specifies the defined action of adding a new user.
Add Windows Hook	Specifies the defined action of adding a new Windows hook.
Add Scheduled Task	Specifies the defined action of adding a scheduled task.
Allocate Virtual Memory in Process	Specifies the defined action of allocating virtual memory in a process.

Bind Address to Socket	Specifies the defined action of binding an address to a socket.
Change Service Configuration	Specifies the defined action of changing the service configuration.
Check for Remote Debugger	Specifies the defined action of checking for a remote debugger.
Close Port	Specifies the defined action of closing a port.
Close Registry Key	Specifies the defined action of closing a registry key.
Close Socket	Specifies the defined action of closing a socket.
Configure Service	Specifies the defined action of configuring a service.
Connect to IP	Specifies the defined action of connecting to an IP address.
Connect to Named Pipe	Specifies the defined action of connecting to a named pipe.
Connect to Network Share	Specifies the defined action of connecting to a network share.
Connect to Socket	Specifies the defined action of connecting to a socket.
Connect to URL	Specifies the defined action of connecting to a URL.
Control Driver	Specifies the defined action of controlling a driver.
Control Service	Specifies the defined action of controlling a service.
Copy File	Specifies the defined action of copying a file.
Create Dialog Box	Specifies the defined action of creating a dialog box.
Create Directory	Specifies the defined action of creating a new directory.
Create Event	Specifies the defined action of creating an event.
Create File	Specifies the defined action of creating a file.
Create File Alternate Data Stream	Specifies the defined action of creating an alternate data stream in a file.

Create File Mapping	Specifies the defined action of creating a new file mapping.
Create File Symbolic Link	Specifies the defined action of creating a file symbolic link.
Create Hidden File	Specifies the defined action of creating a hidden file.
Create Mailslot	Specifies the defined action of creating a mailslot.
Create Module	Specifies the defined action of creating a module.
Create Mutex	Specifies the defined action of creating a mutex.
Create Named Pipe	Specifies the defined action of creating a named pipe.
Create Process	Specifies the defined action of creating a process.
Create Process as User	Specifies the defined action of creating a process as user.
Create Registry Key	Specifies the defined action of creating a registry key.
Create Registry Key Value	Specifies the defined action of creating a registry key value.
Create Remote Thread in Process	Specifies the defined action of creating a remote thread in a process.
Create Service	Specifies the defined action of creating a service.
Create Socket	Specifies the defined action of creating a socket.
Create Symbolic Link	Specifies the defined action of creating a symbolic link.
Create Thread	Specifies the defined action of creating a thread.
Create Window	Specifies the defined action of creating a window.
Delete Directory	Specifies the defined action of deleting a directory.
Delete File	Specifies the defined action of deleting a file.
Delete Named Pipe	Specifies the defined action of deleting a named pipe.
Delete Network Share	Specifies the defined action of deleting a network share.

Delete Registry Key	Specifies the defined action of deleting a registry key.
Delete Registry Key Value	Specifies the defined action of deleting a registry key value.
Delete Service	Specifies the defined action of deleting a service.
Delete User	Specifies the defined action of deleting a user.
Disconnect from Named Pipe	Specifies the defined action of disconnecting from a named pipe.
Disconnect from Network Share	Specifies the defined action of disconnecting from a network share.
Disconnect from Socket	Specifies the defined action of disconnecting from a socket.
Download File	Specifies the defined action of downloading a file.
Enumerate DLLs	Specifies the defined action of enumerating DLLs.
Enumerate Network Shares	Specifies the defined action of enumerating network shares.
Enumerate Protocols	Specifies the defined action of enumerating protocols.
Enumerate Registry Key Subkeys	Specifies the defined action of enumerating registry key subkeys.
Enumerate Registry Key Values	Specifies the defined action of enumerating registry key values.
Enumerate Threads in Process	Specifies the defined action of enumerating threads in a process.
Enumerate Processes	Specifies the defined action of enumerating processes.
Enumerate Services	Specifies the defined action of enumerating services.
Enumerate System Handles	Specifies the defined action of enumerating system handles.
Enumerate Threads	Specifies the defined action of enumerating threads.
Enumerate Users	Specifies the defined action of enumerating users.

Enumerate Windows	Specifies the defined action of enumerating windows.
Find File	Specifies the defined action of finding a file.
Find Window	Specifies the defined action of finding a window.
Flush Process Instruction Cache	Specifies the defined action of flushing the process instruction cache.
Free Library	Specifies the defined action of freeing a library.
Free Process Virtual Memory	Specifies the defined action of freeing virtual memory from a process.
Get Disk Free Space	Specifies the defined action of getting the amount of free space available on a disk.
Get Disk Type	Specifies the defined action of getting the disk type.
Get Elapsed System Up Time	Specifies the defined action of getting the elapsed system up-time.
Get File Attributes	Specifies the defined action of getting file attributes.
Get Function Address	Specifies the defined action of getting the function address.
Get System Global Flags	Specifies the defined action of getting system global flags.
Get Host By Address	Specifies the defined action of getting host by address.
Get Host By Name	Specifies the defined action of getting host by name.
Get Host Name	Specifies the defined action of getting the host name.
Get Library File Name	Specifies the defined action of getting the library file name.
Get Library Handle	Specifies the defined action of getting the library handle.
Get NetBIOS Name	Specifies the defined action of getting the NetBIOS name.
Get Process Current Directory	Specifies the defined action of getting the process's current directory.

Get Process Environment Variable	Specifies the defined action of getting the process environment variable.
Get Process Startup Information	Specifies the defined action of getting the process startup information.
Get Processes Snapshot	Specifies the defined action of getting the processes snapshot.
Get Registry Key Attributes	Specifies the defined action of getting the attributes of a registry key.
Get Service Status	Specifies the defined action of getting the service status.
Get System Global Flags	Specifies the defined action of getting the system global flags.
Get System Local Time	Specifies the defined action of getting the local time on a system.
Get System Host Name	Specifies the defined action of getting the system host name.
Get System NetBIOS Name	Specifies the defined action of getting the NetBIOS name of a system.
Get System Network Parameters	Specifies the defined action of getting the system network parameters.
Get System Time	Specifies the defined action of getting the system time.
Get Thread Context	Specifies the defined action of getting the thread context.
Get Thread Username	Specifies the defined action of getting the thread username.
Get User Attributes	Specifies the defined action of getting the attributes of a user.
Get Username	Specifies the defined action of getting a username.
Get Windows Directory	Specifies the defined action of getting a Windows directory.
Get Windows System Directory	Specifies the defined action of getting a directory.
Get Windows Temporary Files Directory	Specifies the defined action of getting the Windows temporary files directory.

Hide Window	Specifies the defined action of hiding a window.
Impersonate Process	Specifies the defined action of impersonating a process.
Impersonate Thread	Specifies the defined action of impersonating a thread.
Inject Memory Page	Specifies the defined action of injecting a memory page into a process.
Kill Process	Specifies the defined action of killing a process.
Kill Thread	Specifies the defined action of killing a thread.
Kill Window	Specifies the defined action of killing a window.
Listen on Port	Specifies the defined action of listening on a specific port.
Listen on Socket	Specifies the defined action of listening on a socket.
Load and Call Driver	Specifies the defined action of loading and calling a driver.
Load Driver	Specifies the defined action of loading a driver.
Load Library	Specifies the defined action of loading a library.
Load Module	Specifies the defined action of loading a module.
Lock File	Specifies the defined action of locking a file.
Logon as User	Specifies the defined action of logging on as a user.
Map File	Specifies the defined action of mapping a file.
Map Library	Specifies the defined action of mapping a library.
Map View of File	Specifies the defined action of mapping a view of a file.
Modify File	Specifies the defined action of modifying a file.
Modify Named Pipe	Specifies the defined action of modifying a named pipe.
Modify Process	Specifies the defined action of modifying a process.

Modify Service	Specifies the defined action of modifying a service.
Modify Registry Key	Specifies the defined action of modifying a registry key.
Modify Registry Key Value	Specifies the defined action of modifying a registry key value.
Monitor Registry Key	Specifies the defined action of monitoring a registry key.
Move File	Specifies the defined action of moving a file.
Open File	Specifies the defined action of opening a file.
Open File Mapping	Specifies the defined action of opening a file mapping.
Open Mutex	Specifies the defined action of opening a mutex.
Open Port	Specifies the defined action of opening a port.
Open Process	Specifies the defined action of opening a process.
Open Registry Key	Specifies the defined action of opening a registry key.
Open Service	Specifies the defined action of opening a service.
Open Service Control Manager	Specifies the defined action of opening a service control manager.
Protect Virtual Memory	Specifies the defined action of protecting virtual memory.
Query Disk Attributes	Specifies the defined action of querying disk attributes.
Query DNS	Specifies the defined action of querying DNS.
Query Process Virtual Memory	Specifies the defined action of querying process virtual memory.
Queue APC in Thread	Specifies the defined action of querying the asynchronous procedure call (APC) in the context of a thread.
Read File	Specifies the defined action of reading a file.
Read From Named Pipe	Specifies the defined action of reading from a named pipe.

Read From Process Memory	Specifies the defined action of reading from process memory.
Read Registry Key Value	Specifies the defined action of reading a registry key value.
Receive Data on Socket	Specifies the defined action of receiving data on a socket.
Release Mutex	Specifies the defined action of releasing a mutex.
Rename File	Specifies the defined action of renaming a file.
Revert Thread to Self	Specifies the defined action of reverting a thread to its self.
Send Control Code to File	Specifies the defined action of sending a control code to a file.
Send Control Code to Pipe	Specifies the defined action of sending a control code to a pipe.
Send Control Code to Service	Specifies the defined action of sending control code to a service.
Send Data on Socket	Specifies the defined action of sending data on a socket.
Send Data to Address on Socket	Specifies the defined action of sending data to the address on a socket.
Send DNS Query	Specifies the defined action of sending a DNS query.
Send Email Message	Specifies the defined action of sending an email message.
Send ICMP Request	Specifies the defined action of sending an ICMP request.
Send Reverse DNS Query	Specifies the defined action of sending a reverse DNS query.
Set File Attributes	Specifies the defined action of setting file attributes.
Set NetBIOS Name	Specifies the defined action of setting the NetBIOS name.
Set Process Current Directory	Specifies the defined action of setting the process current directory.
Set Process Environment Variable	Specifies the defined action of setting the process environment variable.

Set System Global Flags	Specifies the defined action of setting system global flags.
Set System Host Name	Specifies the defined action of setting the system host name.
Set System Time	Specifies the defined action of setting the system time.
Set Thread Context	Specifies the defined action of setting the thread context.
Show Window	Specifies the defined action of showing a window.
Shutdown System	Specifies the defined action of shutting down a system.
Sleep Process	Specifies the defined action of sleeping a process.
Sleep System	Specifies the defined action of sleeping a system.
Start Service	Specifies the defined action of starting a service.
Unload Driver	Specifies the defined action of unloading a driver.
Unlock File	Specifies the defined action of unlocking a file.
Unmap File	Specifies the defined action of unmapping a file.
Unload Module	Specifies the defined action of unloading a module.
Upload File	Specifies the defined action of uploading a file.
Write to File	Specifies the defined action of writing to a file.
Write to Process Virtual Memory	Specifies the defined action of writing to process virtual memory.

3.4 ActionArgumentNameVocab-1.0 Enumeration

The `ActionArgumentNameVocab` enumeration is the default CybOX vocabulary for Action argument names, captured via the `ActionArgumentType` class (`Argument_Name` property) in CybOX Core. The associated enumeration literals are provided in the table below.

Enumeration Literal	Description
Access Mode	Specifies an argument called access mode.

APC Address	Specifies an argument called APC address.
APC Mode	Specifies an argument called APC mode.
API	Specifies an argument called API.
Application Name	Specifies an argument called application name.
Base Address	Specifies an argument called base address.
Base Address	Specifies an argument called base address.
Callback Address	Specifies an argument called callback address.
Code Address	Specifies an argument called code address.
Command	Specifies an argument called command.
Control Code	Specifies an argument called control code.
Control Parameter	Specifies an argument called control parameter.
Creation Flags	Specifies an argument called creation flags.
Database Name	Specifies an argument called database name.
Delay Time (ms)	Specifies an argument called delay time (ms).
Destination Address	Specifies an argument called destination address.
Error Control	Specifies an argument called initial owner.
File Information Class	Specifies an argument called file information class.
Flags	Specifies an argument called flags.
Function Address	Specifies an argument called function address.
Function Name	Specifies an argument called function name.
Function Name	Specifies an argument called function name.

Function Ordinal	Specifies an argument called function ordinal.
Hook Type	Specifies an argument called hook type.
Host Name	Specifies an argument called host name.
Hostname	Specifies an argument called hostname.
Initial Owner	Specifies an argument called initial owner.
Mapping Offset	Specifies an argument called mapping offset.
Number of Bytes Per Send	Specifies an argument called number of bytes per send.
Options	Specifies an argument called options.
Parameter Address	Specifies an argument called parameter address.
Password	Specifies an argument called password.
Privilege Name	Specifies an argument called privilege name.
Protection	Specifies an argument called protection.
Proxy Bypass	Specifies an argument called proxy bypass.
Proxy Name	Specifies an argument called proxy name.
Reason	Specifies an argument called reason.
Request Size	Specifies an argument called request size.
Requested Version	Specifies an argument called requested version.
Server	Specifies an argument called server.
Service Name	Specifies an argument called service name.
Service State	Specifies an argument called service state.
Service Type	Specifies an argument called service type.

Share Mode	Specifies an argument called share mode.
Shutdown Flag	Specifies an argument called shutdown flag.
Size (bytes)	Specifies an argument called size (bytes).
Sleep Time (ms)	Specifies an argument called sleep time (ms).
Source Address	Specifies an argument called source address.
Starting Address	Specifies an argument called starting address.
System Metric Index	Specifies an argument called system metric index.
Target PID	Specifies an argument called target pid.
Transfer Flags	Specifies an argument called transfer flags.
Username	Specifies an argument called username.

3.5 ActionObjectAssociationTypeVocab-1.0 Enumeration

The `ActionObjectAssocationVocab` enumeration is the default CybOX vocabulary for Action-Object association classes, captured via the `AssociatedObjectType` class (`Association_Type` property) in CybOX Core. The associated enumeration literals are provided in the table below.

Enumeration Literal	Description
Affected	Specifies that the associated object was affected by the action.
Initiating	Specifies that the associated object initiated the action.
Returned	Specifies that the associated object was the result of the action.
Utilized	Specifies that the associated object was utilized by the action.

3.6 ActionRelationshipTypeVocab-1.0 Enumeration

The `ActionObjectAssocationVocab` enumeration is the default CybOX vocabulary for Action-Action relationships, captured via the `ActionRelationshipType` class (`Type` property) in the CybOX Core. The associated enumeration literals are provided in the table below.

Enumeration Literal	Description

Dependent_On	Specifies that this action is dependent on the related action.
Equivalent_To	Specifies that this entity (e.g., Action) is equivalent to the associated entity.
Followed_By	Specifies that this action is followed by the related action.
Initiated	Specifies that this action initiated the related action.
Initiated_By	Specifies that this action was initiated by the related action.
Preceded_By	Specifies that this action is preceded by the related action.
Related_To	Specifies that this action is simply related to the related action in some way.

3.7 EventTypeVocab-1.0.1 Enumeration

The `EventTypeVocab` enumeration is the default CybOX vocabulary for Event classes, captured via the `EventType` class (`Type` property) in the CybOX Core. The associated enumeration literals are provided in the table below.

Enumeration Literal	Description
Account Ops (App Layer)	Specifies the class of events dealing with account operations at the application layer.
Anomaly Events	Specifies the class of events dealing with anomaly events.
API Calls	Specifies the class of events dealing with API calls.
App Layer Traffic	Specifies the class of events dealing with application layer traffic.
Application Logic	Specifies the class of events dealing with application logic.
Authentication Ops	Specifies the class of events dealing with authentication operations.
Authorization (ACL)	Specifies the class of events dealing with authorization via Access Control Lists (ACL).
Autorun	Specifies the class of events dealing with autorun.
Auto-update Ops	Specifies the class of events dealing with auto-update operations.
Basic System Ops	Specifies the class of events dealing with basic system operations.

Configuration Management	Specifies the class of events dealing with configuration management.
Data Flow	Specifies the class of events dealing with data flow.
DHCP	Specifies the class of events dealing with the Dynamic Host Configuration Protocol (DHCP).
DNS Lookup Ops	Specifies the class of events dealing with DNS Lookup operations.
Email Ops	Specifies the class of events dealing with e-mail operations.
File Ops (CRUD)	Specifies the class of events dealing with file operations.
GUI/KVM	Specifies the class of events dealing with the GUI/Kernel-based Virtual Machine (KVM).
HTTP Traffic	Specifies the class of events dealing with HTTP traffic.
IP Ops	Specifies the class of events dealing with IP Operations.
IPC	Specifies the class of events dealing with thread management.
Memory Ops	Specifies the class of events dealing with memory operations.
Packet Traffic	Specifies the class of events dealing with packet traffic.
Port Scan	Specifies the class of events dealing with port scanning.
Privilege Ops	Specifies the class of events dealing with privilege operations.
Procedural Compliance	Specifies the class of events dealing with procedural compliance.
Process Mgt	Specifies the class of events dealing with process management.
Redirection	Specifies the class of events dealing with redirection.
Registry Ops	Specifies the class of events dealing with registry operations.
Service Mgt	Specifies the class of events dealing with service management.
Session Mgt	Specifies the class of events dealing with session management.

Signature Detection	Specifies the class of events dealing with signature detection.
Socket Ops	Specifies the class of events dealing with thread management.
SQL	Specifies the class of events dealing with the SQL language.
Technical Compliance	Specifies the class of events dealing with technical compliance.
Thread Mgt	Specifies the class of events dealing with thread management.
USB/Media Detection	Specifies the class of events dealing with USB and/or media detection.
User/Password Mgt	Specifies the class of events dealing with user/password management.

3.8 EventTypeVocab-1.0 Enumeration

The `EventTypeVocab` enumeration is the default CybOX vocabulary for Event classes, captured via the `EventType` class (`Type` property) in the CybOX Core. The associated enumeration literals are provided in the table below. NOTE: As of CybOX Version 2.1, `EventTypeVocab-1.0` is deprecated. Please use version 1.1 instead (see Section **3.7**).

Enumeration Literal	Description
Account Ops (App Layer)	Specifies the class of events dealing with account operations at the application layer.
Anomoly[1] Events	Specifies the class of events dealing with anomaly events.
API Calls	Specifies the class of events dealing with API calls.
App Layer Traffic	Specifies the class of events dealing with application layer traffic.
Application Logic	Specifies the class of events dealing with application logic.
Authentication Ops	Specifies the class of events dealing with authentication operations.
Authorization (ACL)	Specifies the class of events dealing with authorization via Access Control Lists (ACL).
Autorun	Specifies the class of events dealing with autorun.
Auto-update Ops	Specifies the class of events dealing with auto-update operations.

Basic System Ops	Specifies the class of events dealing with basic system operations.
Configuration Management	Specifies the class of events dealing with configuration management.
Data Flow	Specifies the class of events dealing with data flow.
DHCP	Specifies the class of events dealing with the Dynamic Host Configuration Protocol (DHCP).
DNS Lookup Ops	Specifies the class of events dealing with DNS Lookup operations.
Email Ops	Specifies the class of events dealing with email operations.
File Ops (CRUD)	Specifies the class of events dealing with file operations.
GUI/KVM	Specifies the class of events dealing with the GUI/Kernel-based Virtual Machine (KVM).
HTTP Traffic	Specifies the class of events dealing with HTTP traffic.
IP Ops	Specifies the class of events dealing with IP operations.
IPC	Specifies the class of events dealing with thread management.
Memory Ops	Specifies the class of events dealing with memory operations.
Packet Traffic	Specifies the class of events dealing with packet traffic.
Port Scan	Specifies the class of events dealing with port scanning.
Privilege Ops	Specifies the class of events dealing with privilege operations.
Procedural Compliance	Specifies the class of events dealing with procedural compliance.
Process Mgt	Specifies the class of events dealing with process management.
Redirection	Specifies the class of events dealing with redirection.
Registry Ops	Specifies the class of events dealing with registry operations.
Service Mgt	Specifies the class of events dealing with service management.

Session Mgt	Specifies the class of events dealing with session management.
Signature Detection	Specifies the class of events dealing with signature detection.
Socket Ops	Specifies the class of events dealing with thread management.
SQL	Specifies the class of events dealing with the SQL language.
Technical Compliance	Specifies the class of events dealing with technical compliance.
Thread Mgt	Specifies the class of events dealing with thread management.
USB/Media Detection	Specifies the class of events dealing with USB and/or media detection.
User/Password Mgt	Specifies the class of events dealing with user/password management.

3.9 ObjectRelationshipVocab-1.1 Enumeration

The `ObjectRelationshipVocab` enumeration is the default CybOX vocabulary for Object-Object relationships, captured via the `RelatedObjectType` class (`Relationship` property) in CybOX Core. The associated enumeration literals are provided in the table below.

Enumeration Literal	Description
Allocated	Specifies that this object allocated the related object.
Allocated_By	Specifies that this object was allocated by the related object.
Bound	Specifies that this object bound the related object.
Bound_By	Specifies that this object was bound by the related object.
Characterized_By	Specifies that the related object describes the properties of this object. This is most applicable in cases where the related object is a non-Artifact Object and this object is an Artifact Object.
Characterizes	Specifies that this object describes the properties of the related object. This is most applicable in cases where the related object is an Artifact Object and this object is a non-Artifact Object.
Child_Of	Specifies that this object is a child of the related object.
Closed	Specifies that this object closed the related object.

Closed_By	Specifies that this object was closed by the related object.
Compressed	Specifies that this object compressed the related object.
Compressed_By	Specifies that this object was compressed by the related object.
Compressed_From	Specifies that this object was compressed from the related object.
Compressed_Into	Specifies that this object was compressed into the related object.
Connected_From	Specifies that this object was connected to from the related object.
Connected_To	Specifies that this object connected to the related object.
Contained_Within	Specifies that this object is contained within the related object.
Contains	Specifies that this object contains the related object.
Copied	Specifies that this object copied the related object.
Copied_By	Specifies that this object was copied by the related object.
Copied_From	Specifies that this object was copied from the related object.
Copied_To	Specifies that this object was copied to the related object.
Created	Specifies that this object created the related object.
Created_By	Specifies that this object was created by the related object.
Decoded	Specifies that this object decoded the related object.
Decoded_By	Specifies that this object was decoded by the related object.
Decompressed	Specifies that this object decompressed the related object.
Decompressed_By	Specifies that this object was decompressed by the related object.
Decrypted	Specifies that this object decrypted the related object.
Decrypted_By	Specifies that this object was decrypted by the related object.

Deleted	Specifies that this object deleted the related object.
Deleted_By	Specifies that this object was deleted by the related object.
Deleted_From	Specifies that this object was deleted from the related object.
Downloaded	Specifies that this object downloaded the related object.
Downloaded_By	Specifies that this object was downloaded by the related object.
Downloaded_From	Specifies that this object was downloaded from the related object.
Downloaded_To	Specifies that this object downloaded the related object.
Dropped	Specifies that this object dropped the related object.
Dropped_By	Specifies that this object was dropped by the related object.
Encoded	Specifies that this object encoded the related object.
Encoded_By	Specifies that this object was encoded by the related object.
Encrypted	Specifies that this object encrypted the related object.
Encrypted_By	Specifies that this object was encrypted by the related object.
Encrypted_From	Specifies that this object was encrypted from the related object.
Encrypted_To	Specifies that this object was encrypted to the related object.
Extracted_From	Specifies that this object was extracted from the related object.
FQDN_Of	Specifies that this object is an FQDN of the related object.
Freed	Specifies that this object freed the related object.
Freed_By	Specifies that this object was freed by the related object.
Hooked	Specifies that this object hooked the related object.
Hooked_By	Specifies that this object was hooked by the related object.

Initialized_By	Specifies that this object was initialized by the related object.
Initialized_To	Specifies that this object was initialized to the related object.
Injected	Specifies that this object injected the related object.
Injected_As	Specifies that this object injected as the related object.
Injected_By	Specifies that this object was injected by the related object.
Injected_Into	Specifies that this object injected into the related object.
Installed	Specifies that this object installed the related object.
Installed_By	Specifies that this object was installed by the related object.
Joined	Specifies that this object joined the related object.
Joined_By	Specifies that this object was joined by the related object.
Killed	Specifies that this object killed the related object.
Killed_By	Specifies that this object was killed by the related object.
Listened_On	Specifies that this object listened on the related object.
Listened_On_By	Specifies that this object was listened on by the related object.
Loaded_From	Specifies that this object was loaded from the related object.
Loaded_Into	Specifies that this object loaded into the related object.
Locked	Specifies that this object locked the related object.
Locked_By	Specifies that this object was locked by the related object.
Mapped_By	Specifies that this object was mapped by the related object.
Mapped_Into	Specifies that this object was mapped into the related object.
Merged	Specifies that this object merged the related object.

Merged_By	Specifies that this object was merged by the related object.
Modified_Properties_Of	Specifies that this object modified the properties of the related object.
Monitored	Specifies that this object monitored the related object.
Monitored_By	Specifies that this object was monitored by the related object.
Moved	Specifies that this object moved the related object.
Moved_By	Specifies that this object was moved by the related object.
Moved_From	Specifies that this object was moved from the related object.
Moved_To	Specifies that this object was moved to the related object.
Opened	Specifies that this object opened the related object.
Opened_By	Specifies that this object was opened by the related object.
Packed	Specifies that this object packed the related object.
Packed_By	Specifies that this object was packed by the related object.
Packed_From	Specifies that this object was packed from the related object.
Packed_Into	Specifies that this object was packed into the related object.
Parent_Of	Specifies that this object is a parent of the related object.
Paused	Specifies that this object paused the related object.
Paused_By	Specifies that this object was paused by the related object.
Previously_Contained	Specifies that this object previously contained the related object.
Properties_Modified_By	Specifies that the properties of this object were modified by the related object.
Properties_Queried	Specifies that the object queried properties of the related object.
Properties_Queried_By	Specifies that the properties of this object were queried by the related

	object.
Read_From	Specifies that this object was read from the related object.
Read_From_By	Specifies that this object was read from by the related object.
Received	Specifies that this object received the related object.
Received_By	Specifies that this object was received by the related object.
Received_From	Specifies that this object was received from the related object.
Received_Via_Upload	Specifies that this object received the related object via upload.
Redirects_To	Specifies that this object redirects to the related object.
Related_To	Specifies that this object is related to the related object.
Renamed	Specifies that this object renamed the related object.
Renamed_By	Specifies that this object was renamed by the related object.
Renamed_From	Specifies that this object was renamed from the related object.
Renamed_To	Specifies that this object was renamed to the related object.
Resolved_To	Specifies that this object was resolved to the related object.
Resumed	Specifies that this object resumed the related object.
Resumed_By	Specifies that this object was resumed by the related object.
Root_Domain_Of	Specifies that this object is the root domain of the related object.
Searched_For	Specifies that this object searched for the related object.
Searched_For_By	Specifies that this object was searched for by the related object.
Sent	Specifies that this object sent the related object.
Sent_By	Specifies that this object was sent by the related object.

Sent_To	Specifies that this object was sent to the related object.
Sent_Via_Upload	Specifies that this object sent the related object via upload.
Set_From	Specifies that this object was set from the related object.
Set_To	Specifies that this object was set to the related object.
Sub-domain_Of	Specifies that this object is a sub-domain of the related object.
Supra-domain_Of	Specifies that this object is a supra-domain of the related object.
Suspended	Specifies that this object suspended the related object.
Suspended_By	Specifies that this object was suspended by the related object.
Unhooked	Specifies that this object unhooked the related object.
Unhooked_By	Specifies that this object was unhooked by the related object.
Unlocked	Specifies that this object unlocked the related object.
Unlocked_By	Specifies that this object was unlocked by the related object.
Unpacked	Specifies that this object unpacked the related object.
Unpacked_By	Specifies that this object was unpacked by the related object.
Uploaded	Specifies that this object uploaded the related object.
Uploaded_By	Specifies that this object was uploaded by the related object.
Uploaded_From	Specifies that this object was uploaded from the related object.
Uploaded_To	Specifies that this object was uploaded to the related object.
Used	Specifies that this object used the related object.
Used_By	Specifies that this object was used by the related object.
Values_Enumerated	Specifies that the object enumerated values of the related object.

Values_Enumerated_By	Specifies that the values of the object were enumerated by the related object.
Written_To_By	Specifies that this object was written to by the related object.
Wrote_To	Specifies that this object wrote to the related object.

3.10 ObjectRelationshipVocab-1.0 Enumeration

The `ObjectRelationshipVocab` enumeration is the default CybOX vocabulary for Object-Object relationships, captured via the `RelatedObjectType` class (`Relationship` property) in CybOX Core. The associated enumeration literals are provided in the table below. NOTE: As of CybOX Version 2.1, `ObjectRelationshipVocab-1.0` is deprecated. Please use version 1.1 instead (see Section **3.9**).

Enumeration Literal	Description
Allocated	Specifies that this object allocated the related object.
Allocated_By	Specifies that this object was allocated by the related object.
Bound	Specifies that this object bound the related object.
Bound_By	Specifies that this object was bound by the related object.
Characterized_By	Specifies that the related object describes the properties of this object. This is most applicable in cases where the related object is a non-artifact object and this object is an artifact object.
Characterizes	Specifies that this object describes the properties of the related object. This is most applicable in cases where the related object is an Artifact object and this object is a non-artifact object.
Child_Of	Specifies that this object is a child of the related object.
Closed	Specifies that this object closed the related object.
Closed_By	Specifies that this object was closed by the related object.
Compressed	Specifies that this object compressed the related object.
Compressed_By	Specifies that this object was compressed by the related object.
Compressed_From	Specifies that this object was compressed from the related object.

Compressed_Into	Specifies that this object was compressed into the related object.
Connected_From	Specifies that this object was connected to from the related object.
Connected_To	Specifies that this object connected to the related object.
Contained_Within	Specifies that this object is contained within the related object.
Contains	Specifies that this object contains the related object.
Copied	Specifies that this object copied the related object.
Copied_By	Specifies that this object was copied by the related object.
Copied_From	Specifies that this object was copied from the related object.
Copied_To	Specifies that this object was copied to the related object.
Created	Specifies that this object created the related object.
Created_By	Specifies that this object was created by the related object.
Decoded	Specifies that this object decoded the related object.
Decoded_By	Specifies that this object was decoded by the related object.
Decompressed	Specifies that this object decompressed the related object.
Decompressed_By	Specifies that this object was decompressed by the related object.
Decrypted	Specifies that this object decrypted the related object.
Decrypted_By	Specifies that this object was decrypted by the related object.
Deleted	Specifies that this object deleted the related object.
Deleted_By	Specifies that this object was deleted by the related object.
Deleted_From	Specifies that this object was deleted from the related object.
Downloaded	Specifies that this object downloaded the related object.

Downloaded_By	Specifies that this object was downloaded by the related object.
Downloaded_From	Specifies that this object was downloaded from the related object.
Downloaded_To	Specifies that this object downloaded the related object.
Dropped	Specifies that this object dropped the related object.
Dropped_By	Specifies that this object was dropped by the related object.
Encoded	Specifies that this object encoded the related object.
Encoded_By	Specifies that this object was encoded by the related object.
Encrypted	Specifies that this object encrypted the related object.
Encrypted_By	Specifies that this object was encrypted by the related object.
Encrypted_From	Specifies that this object was encrypted from the related object.
Encrypted_To	Specifies that this object was encrypted to the related object.
Extracted_From	Specifies that this object was extracted from the related object.
FQDN_Of	Specifies that this object is an FQDN of the related object.
Freed	Specifies that this object freed the related object.
Freed_By	Specifies that this object was freed by the related object.
Hooked	Specifies that this object hooked the related object.
Hooked_By	Specifies that this object was hooked by the related object.
Initialized_By	Specifies that this object was initialized by the related object.
Initialized_To	Specifies that this object was initialized to the related object.
Injected	Specifies that this object injected the related object.
Injected_As	Specifies that this object injected as the related object.

Injected_By	Specifies that this object was injected by the related object.
Injected_Into	Specifies that this object injected into the related object.
Installed	Specifies that this object installed the related object.
Installed_By	Specifies that this object was installed by the related object.
Joined	Specifies that this object joined the related object.
Joined_By	Specifies that this object was joined by the related object.
Killed	Specifies that this object killed the related object.
Killed_By	Specifies that this object was killed by the related object.
Listened_On	Specifies that this object listened on the related object.
Listened_On_By	Specifies that this object was listened on by the related object.
Loaded_From	Specifies that this object was loaded from the related object.
Loaded_Into	Specifies that this object loaded into the related object.
Locked	Specifies that this object locked the related object.
Locked_By	Specifies that this object was locked by the related object.
Mapped_By	Specifies that this object was mapped by the related object.
Mapped_Into	Specifies that this object was mapped into the related object.
Merged	Specifies that this object merged the related object.
Merged_By	Specifies that this object was merged by the related object.
Modified_Properties_Of	Specifies that this object modified the properties of the related object.
Monitored	Specifies that this object monitored the related object.
Monitored_By	Specifies that this object was monitored by the related object.

Moved	Specifies that this object moved the related object.
Moved_By	Specifies that this object was moved by the related object.
Moved_From	Specifies that this object was moved from the related object.
Moved_To	Specifies that this object was moved to the related object.
Opened	Specifies that this object opened the related object.
Opened_By	Specifies that this object was opened by the related object.
Packed	Specifies that this object packed the related object.
Packed_By	Specifies that this object was packed by the related object.
Packed_From	Specifies that this object was packed from the related object.
Packed_Into	Specifies that this object was packed into the related object.
Parent_Of	Specifies that this object is a parent of the related object.
Paused	Specifies that this object paused the related object.
Paused_By	Specifies that this object was paused by the related object.
Previously_Contained	Specifies that this object previously contained the related object.
Properties_Modified_By	Specifies that the properties of this object were modified by the related object.
Properties_Queried	Specifies that the object queried properties of the related object.
Properties_Queried_By	Specifies that the properties of this object were queried by the related object.
Read_From	Specifies that this object was read from the related object.
Read_From_By	Specifies that this object was read from by the related object.
Received	Specifies that this object received the related object.

Received_By	Specifies that this object was received by the related object.
Received_From	Specifies that this object was received from the related object.
Received_Via_Upload	Specifies that this object received the related object via upload.
Related_To	Specifies that this object is related to the related object.
Renamed	Specifies that this object renamed the related object.
Renamed_By	Specifies that this object was renamed by the related object.
Renamed_From	Specifies that this object was renamed from the related object.
Renamed_To	Specifies that this object was renamed to the related object.
Resolved_To	Specifies that this object was resolved to the related object.
Resumed	Specifies that this object resumed the related object.
Resumed_By	Specifies that this object was resumed by the related object.
Root_Domain_Of	Specifies that this object is the root domain of the related object.
Searched_For	Specifies that this object searched for the related object.
Searched_For_By	Specifies that this object was searched for by the related object.
Sent	Specifies that this object sent the related object.
Sent_By	Specifies that this object was sent by the related object.
Sent_To	Specifies that this object was sent to the related object.
Sent_Via_Upload	Specifies that this object sent the related object via upload.
Set_From	Specifies that this object was set from the related object.
Set_To	Specifies that this object was set to the related object.
Sub-domain_Of	Specifies that this object is a sub-domain of the related object.

Supra-domain_Of	Specifies that this object is a supra-domain of the related object.
Suspended	Specifies that this object suspended the related object.
Suspended_By	Specifies that this object was suspended by the related object.
Unhooked	Specifies that this object unhooked the related object.
Unhooked_By	Specifies that this object was unhooked by the related object.
Unlocked	Specifies that this object unlocked the related object.
Unlocked_By	Specifies that this object was unlocked by the related object.
Unpacked	Specifies that this object unpacked the related object.
Unpacked_By	Specifies that this object was unpacked by the related object.
Uploaded	Specifies that this object uploaded the related object.
Uploaded_By	Specifies that this object was uploaded by the related object.
Uploaded_From	Specifies that this object was uploaded from the related object.
Uploaded_To	Specifies that this object was uploaded to the related object.
Values_Enumerated	Specifies that the object enumerated values of the related object.
Values_Enumerated_By	Specifies that the values of the object were enumerated by the related object.
Written_To_By	Specifies that this object was written to by the related object.
Wrote_To	Specifies that this object wrote to the related object.

3.11 ObjectStateVocab-1.0 Enumeration

The `ObjectStateVocab` enumeration is the default CybOX vocabulary for Object states, captured via the `ObjectType` class (`State` property) in CybOX Core. The associated enumeration literals are provided in the table below.

Enumeration Literal	Description

Active	Specifies that the object is active.
Closed	Specifies that the object is closed.
Does Not Exist	Specifies that the object does not exist.
Exists	Specifies that the object exists.
Inactive	Specifies that the object is inactive.
Locked	Specifies that the object is locked.
Open	Specifies that the object is open.
Started	Specifies that the object has started.
Stopped	Specifies that the object has stopped.
Unlocked	Specifies that the object is unlocked.

3.12 CharacterEncodingVocab-1.0 Enumeration

The `CharacterEncodingVocab` enumeration is the default CybOX vocabulary for character encoding, used in the `ExtractedStringType` class (`Encoding` property) in CybOX Common. The associated enumeration literals are provided in the table below.

Enumeration Literal	Description
ASCII	Specifies the American Standard Code for Information Interchange (ASCII) character encoding scheme.
UTF-8	Specifies the UCS Transformation Format-8 bit (UTF-8) character encoding scheme.
UTF-16	Specifies the UCS Transformation Format-16 bit (UTF-16) character encoding scheme.
UTF-32	Specifies the UCS Transformation Format-32 bit (UTF-32) character encoding scheme.
Windows-1250	Specifies the Windows-1250 character encoding scheme, for Central European languages.

Windows-1251	Specifies the Windows-1251 character encoding scheme, for Cyrillic alphabets.
Windows-1252	Specifies the Windows-1252 character encoding scheme, for Western languages.
Windows-1253	Specifies the Windows-1253 character encoding scheme, for Greek.
Windows-1254	Specifies the Windows-1254 character encoding scheme, for Turkish.
Windows-1255	Specifies the Windows-1255 character encoding scheme, for Hebrew.
Windows-1256	Specifies the Windows-1256 character encoding scheme, for Arabic.
Windows-1257	Specifies the Windows-1257 character encoding scheme, for Baltic languages.
Windows-1258	Specifies the Windows-1258 character encoding scheme, for Vietnamese.

3.13 InformationSourceTypeVocab-1.0 Enumeration

The `InformationSourceTypeVocab` enumeration is the default CybOX vocabulary for information source classes, used in the `MeasureSourceType` class (`Information_Source_Type` property) in CybOX Common. The associated enumeration literals are provided in the table below.

Enumeration Literal	Description
Application Framework	Specifies a cyber observation coming from an application framework.
Application Logs	Specifies a cyber observation coming from application logs.
Comm Logs	Specifies a cyber observation coming from communications logs.
DBMS Log	Specifies a cyber observation coming from the DBMS log.
Frameworks	Specifies a cyber observation coming from frameworks.
Help Desk	Specifies a cyber observation coming from a human or automated help desk.
IAVM	Specifies a cyber observation made using information provided by The information assurance vulnerability management (IAVM) mechanisms.
Incident Management	Specifies a cyber observation made using information provided by Incident Management services.

OS/Device Driver APIs	Specifies a cyber observation coming from OS/Device Driver APIs.
TPM	Specifies a cyber observation made using TPM output data.
VM Hypervisor	Specifies a cyber observation coming from the VM hypervisor data.
Web Logs	Specifies a cyber observation coming from web logs.

3.14 HashNameVocab-1.0 Enumeration

The `HashNameVocab` enumeration is the default CybOX vocabulary for hashing algorithm names, used in the `HashType` class (`Type` property) in CybOX Common. The associated enumeration literals are provided in the table below.

Enumeration Literal	Description
MD5	The MD5 value specifies the MD5 hashing algorithm.
MD6	The MD6 value specifies the MD6 hashing algorithm.
SHA1	The SHA1 value specifies the SHA1 hashing algorithm.
SHA224	The SHA24 value specifies the SHA224 hashing algorithm.
SHA256	The SHA256 value specifies the SHA256 hashing algorithm.
SHA384	The SHA384 value specifies the SHA384 hashing algorithm.
SHA512	The SHA512 value specifies the SHA512 hashing algorithm.
SSDEEP	The SSDEEP value specifies the SSDEEP hashing algorithm.

3.15 ToolTypeVocab-1.1 Enumeration

The `ToolTypeVocab` enumeration is the default CybOX vocabulary for tool classes, used in the `ToolInformationType` class (`Type` property) in CybOX Common. The associated enumeration literals are provided in the table below.

Enumeration Literal	Description
Asset Scanner	Specifies an asset scanner tool.
AV	Specifies an anti-virus (AV) tools and/or software.

Configuration Scanner	Specifies a configuration scanner tool.
DBMS Monitor	Specifies a Database Management System (DBMS) monitor tool.
Digital Forensics	Specifies a digital forensics tool.
Dynamic Malware Analysis	Specifies a dynamic malware Analysis tool.
Firewall	Specifies a software or hardware firewall.
Gateway	Specifies a cyber observation made using a software or hardware network gateway.
HIDS	Specifies a Host-based Intrusion Detection System (HIDS) tool.
HIPS	Specifies a Host-based Intrusion Protection System (HIPS) tool.
Intelligence Service Platform	Specifies an intelligence service platform tool.
Network Configuration Management Tool	Specifies a network configuration management tool.
Network Flow Capture and Analysis	Specifies a network flow capture and analysis tool.
NIDS	Specifies a Network Intrusion Detection System (NIDS) tool.
NIPS	Specifies a Network Intrusion Protection System (NIPS) tool.
Packet Capture and Analysis	Specifies a packet capture and analysis tool.
Proxy	Specifies a cyber observation made using a software or hardware network proxy.
Router	Specifies a software or hardware router.
SEM	Specifies a Security Event Management (SEM) tool.
SIM	Specifies a Security Information Management (SIM) tool.

SNMP/MIBs	Specifies a SNMP or MIBs (Simple Network Management Protocol or Management Information Base) tool.
Static Malware Analysis	Specifies a static malware analysis tool.
System Configuration Management Tool	Specifies a system configuration management tool.
Vulnerability Scanner	The vulnerability scanner value specifies a vulnerability scanner tool.

3.16 ToolTypeVocab-1.0 Enumeration

The `ToolTypeVocab` enumeration is the default CybOX vocabulary for tool class, used in the `ToolInformationType` class (`Type` property) in CybOX Common. The associated enumeration literals are provided in the table below. NOTE: As of CybOX Version 2.1, `ToolTypeVocab-1.0` is deprecated. Please use version 1.1 instead (see Section **3.15**).

Enumeration Literal	Description
A/V	Specifies an anti-virus (AV) tools and/or software.
Asset Scanner	Specifies an asset scanner tool.
Configuration Scanner	Specifies a configuration scanner tool.
DBMS Monitor	Specifies a Database Management System (DBMS) monitor tool.
Firewall	Specifies a software or hardware firewall.
Gateway	Specifies a cyber observation made using a software or hardware network gateway.
HIDS	Specifies a Host-based Intrusion Detection System (HIDS) tool.
HIPS	Specifies a Host-based Intrusion Protection System (HIPS) tool.
NIDS	Specifies a Network Intrusion Detection System (NIDS) tool.
NIPS	Specifies a Network Intrusion Protection System (NIPS) tool.
Proxy	The proxy value specifies a cyber observation made using a network proxy.

Router	The router value specifies a cyber observation made using a router.
SEM	Specifies a Security Event Management (SEM) tool.
SIM	Specifies a Security Information Management (SIM) tool.
SNMP/MIBs	Specifies a SNMP or MIBs (Simple Network Management Protocol or Management Information Base) tool.
Vulnerability Scanner	The vulnerability scanner value specifies a cyber observation made using a vulnerability scanner.

4 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, 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 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, 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.

Appendix A. Acknowledgments

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

Aetna
David Crawford
AIT Austrian Institute of Technology
Roman Fiedler
Florian Skopik
Australia and New Zealand Banking Group (ANZ Bank)
Dean Thompson
Blue Coat Systems, Inc.
Owen Johnson
Bret Jordan
Century Link
Cory Kennedy
CIRCL
Alexandre Dulaunoy
Andras Iklody
Raphaël Vinot
Citrix Systems
Joey Peloquin
Dell
Will Urbanski
Jeff Williams
DTCC
Dan Brown
Gordon Hundley
Chris Koutras
EMC
Robert Griffin
Jeff Odom
Ravi Sharda
Financial Services Information Sharing and Analysis Center (FS-ISAC)
David Eilken
Chris Ricard
Fortinet Inc.
Gavin Chow
Kenichi Terashita
Airbus Group SAS
Joerg Eschweiler
Marcos Orallo
Anomali
Ryan Clough
Wei Huang
Hugh Njemanze
Katie Pelusi
Aaron Shelmire
Jason Trost
Bank of America
Alexander Foley
Center for Internet Security (CIS)
Sarah Kelley
Check Point Software Technologies
Ron Davidson
Cisco Systems
Syam Appala
Ted Bedwell
David McGrew
Pavan Reddy
Omar Santos
Jyoti Verma
Cyber Threat Intelligence Network, Inc. (CTIN)
Doug DePeppe
Jane Ginn
Ben Othman
DHS Office of Cybersecurity and Communications (CS&C)
Richard Struse
Marlon Taylor
EclecticIQ
Marko Dragoljevic
Joep Gommers
Sergey Polzunov
Rutger Prins

Fujitsu Limited
Neil Edwards
Frederick Hirsch
Ryusuke Masuoka
Daisuke Murabayashi
Google Inc.
Mark Risher
Hitachi, Ltd.
Kazuo Noguchi
Akihito Sawada
Masato Terada
iboss, Inc.
Paul Martini
Individual
Jerome Athias
Peter Brown
Elysa Jones
Sanjiv Kalkar
Bar Lockwood
Terry MacDonald
Alex Pinto
Intel Corporation
Tim Casey
Kent Landfield
JPMorgan Chase Bank, N.A.
Terrence Driscoll
David Laurance
LookingGlass
Allan Thomson
Lee Vorthman
Mitre Corporation
Greg Back
Jonathan Baker
Sean Barnum
Desiree Beck
Nicole Gong
Jasen Jacobsen
Ivan Kirillov
Richard Piazza
Jon Salwen
Charles Schmidt
Andrei Sîrghi
Raymon van der Velde
eSentire, Inc.
Jacob Gajek
FireEye, Inc.
Phillip Boles
Pavan Gorakav
Anuj Kumar
Shyamal Pandya
Paul Patrick
Scott Shreve
Fox-IT
Sarah Brown
Georgetown University
Eric Burger
Hewlett Packard Enterprise (HPE)
Tomas Sander
IBM
Peter Allor
Eldan Ben-Haim
Sandra Hernandez
Jason Keirstead
John Morris
Laura Rusu
Ron Williams
IID
Chris Richardson
Integrated Networking Technologies, Inc.
Patrick Maroney
Johns Hopkins University Applied Physics Laboratory
Karin Marr
Julie Modlin
Mark Moss
Pamela Smith
Kaiser Permanente
Russell Culpepper
Beth Pumo
Lumeta Corporation
Brandon Hoffman
MTG Management Consultants, LLC.
James Cabral

Emmanuelle Vargas-Gonzalez
John Wunder

National Council of ISACs (NCI)
Scott Algeier
Denise Anderson
Josh Poster

NEC Corporation
Takahiro Kakumaru

North American Energy Standards Board
David Darnell

Object Management Group
Cory Casanave

Palo Alto Networks
Vishaal Hariprasad

Queralt, Inc.
John Tolbert

Resilient Systems, Inc.
Ted Julian

Securonix
Igor Baikalov

Siemens AG
Bernd Grobauer

Soltra
John Anderson
Aishwarya Asok Kumar
Peter Ayasse
Jeff Beekman
Michael Butt
Cynthia Camacho
Aharon Chernin
Mark Clancy
Brady Cotton
Trey Darley
Mark Davidson
Paul Dion
Daniel Dye
Robert Hutto
Raymond Keckler
Ali Khan
Chris Kiehl
Clayton Long

National Security Agency
Mike Boyle
Jessica Fitzgerald-McKay

New Context Services, Inc.
John-Mark Gurney
Christian Hunt
James Moler
Daniel Riedel
Andrew Storms

OASIS
James Bryce Clark
Robin Cover
Chet Ensign

Open Identity Exchange
Don Thibeau

PhishMe Inc.
Josh Larkins

Raytheon Company-SAS
Daniel Wyschogrod

Retail Cyber Intelligence Sharing Center (R-CISC)
Brian Engle

Semper Fortis Solutions
Joseph Brand

Splunk Inc.
Cedric LeRoux
Brian Luger
Kathy Wang

TELUS
Greg Reaume
Alan Steer

Threat Intelligence Pty Ltd
Tyron Miller
Andrew van der Stock

ThreatConnect, Inc.
Wade Baker
Cole Iliff
Andrew Pendergast
Ben Schmoker
Jason Spies

TruSTAR Technology
Chris Roblee

Michael Pepin
Natalie Suarez
David Waters
Benjamin Yates

Symantec Corp.

Curtis Kostrosky

The Boeing Company

Crystal Hayes

ThreatQuotient, Inc.

Ryan Trost

U.S. Bank

Mark Angel
Brad Butts
Brian Fay
Mona Magathan
Yevgen Sautin

US Department of Defense (DoD)

James Bohling
Eoghan Casey
Gary Katz
Jeffrey Mates

VeriSign

Robert Coderre
Kyle Maxwell
Eric Osterweil

United Kingdom Cabinet Office

Iain Brown
Adam Cooper
Mike McLellan
Chris O'Brien
James Penman
Howard Staple
Chris Taylor
Laurie Thomson
Alastair Treharne
Julian White
Bethany Yates

US Department of Homeland Security

Evette Maynard-Noel
Justin Stekervetz

ViaSat, Inc.

Lee Chieffalo
Wilson Figueroa
Andrew May

Yaana Technologies, LLC

Anthony Rutkowski

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

Appendix B. Revision History

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

[1] Corrected in `EventTypeVocab-1.0.1`