CybOXTM Version 2.1.1 Part 43: PDF File Object

Working Draft 01

16 November 2015

Technical Committee:

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

Chair:

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

Editors:

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

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

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

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

Additional artifacts:

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

* *CybOXTM Version 2.1.1 Part 1: Overview*. [URI]
* *CybOXTM Version 2.1.1 Part 2: Common*. [URI]
* *CybOXTM Version 2.1.1 Part 3: Core*. [URI]
* *CybOXTM Version 2.1.1 Part 4: Default Extensions*. [URI]
* *CybOXTM Version 2.1.1 Part 5: Vocabularies*. [URI]
* *CybOXTM Version 2.1.1 Part 6: UML Model*. [URI]
* *CybOXTM Version 2.1.1 Part 7: API Object*. [URI]
* *CybOXTM Version 2.1.1 Part 8: ARP Cache Object*. [URI]
* *CybOXTM Version 2.1.1 Part 9: AS Object*. [URI]
* *CybOXTM Version 2.1.1 Part 10: Account Object*. [URI]
* *CybOXTM Version 2.1.1 Part 11: Address Object*. [URI]
* *CybOXTM Version 2.1.1 Part 12: Archive File Object*. [URI]
* *CybOXTM Version 2.1.1 Part 13: Artifact Object*. [URI]
* *CybOXTM Version 2.1.1 Part 14: Code Object*. [URI]
* *CybOXTM Version 2.1.1 Part 15: Custom Object*. [URI]
* *CybOXTM Version 2.1.1 Part 16: DNS Cache Object*. [URI]
* *CybOXTM Version 2.1.1 Part 17: DNS Query Object*. [URI]
* *CybOXTM Version 2.1.1 Part 18: DNS Record Object*. [URI]
* *CybOXTM Version 2.1.1 Part 19: Device Object*. [URI]
* *CybOXTM Version 2.1.1 Part 20: Disk Object*. [URI]
* *CybOXTM Version 2.1.1 Part 21: Disk Partition Object*. [URI]
* *CybOXTM Version 2.1.1 Part 22: Domain Name Object*. [URI]
* *CybOXTM Version 2.1.1 Part 23: Email Message Object*. [URI]
* *CybOXTM Version 2.1.1 Part 24: File Object*. [URI]
* *CybOXTM Version 2.1.1 Part 25: GUI Dialogbox Object*. [URI]
* *CybOXTM Version 2.1.1 Part 26: GUI Object*. [URI]
* *CybOXTM Version 2.1.1 Part 27: GUI Window Object*. [URI]
* *CybOXTM Version 2.1.1 Part 28: HTTP Session Object*. [URI]
* *CybOXTM Version 2.1.1 Part 29: Hostname Session Object*. [URI]
* *CybOXTM Version 2.1.1 Part 30: Image File Object*. [URI]
* *CybOXTM Version 2.1.1 Part 31: Library File Object*. [URI]
* *CybOXTM Version 2.1.1 Part 32: Link Object*. [URI]
* *CybOXTM Version 2.1.1 Part 33: Linux Package Object*. [URI]
* *CybOXTM Version 2.1.1 Part 34: Memory Object*. [URI]
* *CybOXTM Version 2.1.1 Part 35: Mutex Object*. [URI]
* *CybOXTM Version 2.1.1 Part 36: Network Connection Object*. [URI]
* *CybOXTM Version 2.1.1 Part 37: Network Flow Object*. [URI]
* *CybOXTM Version 2.1.1 Part 38: Network Packet Object*. [URI]
* *CybOXTM Version 2.1.1 Part 39: Network Route Entry Object*. [URI]
* *CybOXTM Version 2.1.1 Part 40: Network Route Object*. [URI]
* *CybOXTM Version 2.1.1 Part 41: Network Socket Object*. [URI]
* *CybOXTM Version 2.1.1 Part 42: Network Subnet Object*. [URI]
* *CybOXTM Version 2.1.1 Part 43: PDF File Object*. (this document)
* *CybOXTM Version 2.1.1 Part 44: Pipe Object*. [URI]
* *CybOXTM Version 2.1.1 Part 45: Port Object*. [URI]
* *CybOXTM Version 2.1.1 Part 46: Process Object*. [URI]
* *CybOXTM Version 2.1.1 Part 47: Product Object*. [URI]
* *CybOXTM Version 2.1.1 Part 48: SMS Message Object*. [URI]
* *CybOXTM Version 2.1.1 Part 49: Semaphore Object*. [URI]
* *CybOXTM Version 2.1.1 Part 50: Socket Address Object*. [URI]
* *CybOXTM Version 2.1.1 Part 51: System Object*. [URI]
* *CybOXTM Version 2.1.1 Part 52: URI Object*. [URI]
* *CybOXTM Version 2.1.1 Part 53: URL History Object*. [URI]
* *CybOXTM Version 2.1.1 Part 54: Unix File Object*. [URI]
* *CybOXTM Version 2.1.1 Part 55: Unix Network Route Entry Object*. [URI]
* *CybOXTM Version 2.1.1 Part 56: Unix Pipe Object*. [URI]
* *CybOXTM Version 2.1.1 Part 57: Unix Process Object*. [URI]
* *CybOXTM Version 2.1.1 Part 58: Unix User Account Object*. [URI]
* *CybOXTM Version 2.1.1 Part 59: Unix Volume Object*. [URI]
* *CybOXTM Version 2.1.1 Part 60: Unix Account Object*. [URI]
* *CybOXTM Version 2.1.1 Part 61: User Session Object*. [URI]
* *CybOXTM Version 2.1.1 Part 62: Volume Object*. [URI]
* *CybOXTM Version 2.1.1 Part 63: Whois Object*. [URI]
* *CybOXTM Version 2.1.1 Part 64: Win Computer Account Object*. [URI]
* *CybOXTM Version 2.1.1 Part 65: Win Critical Section Object*. [URI]
* *CybOXTM Version 2.1.1 Part 66: Win Driver Object*. [URI]
* *CybOXTM Version 2.1.1 Part 67: Win Event Log Object*. [URI]
* *CybOXTM Version 2.1.1 Part 68: Win Event Object*. [URI]
* *CybOXTM Version 2.1.1 Part 69: Win Executable File Object*. [URI]
* *CybOXTM Version 2.1.1 Part 70: Win File Object*. [URI]
* *CybOXTM Version 2.1.1 Part 71: Win Filemapping Object*. [URI]
* *CybOXTM Version 2.1.1 Part 72: Win Handle Object*. [URI]
* *CybOXTM Version 2.1.1 Part 73: Win Hook Object*. [URI]
* *CybOXTM Version 2.1.1 Part 74: Win Kernel Hook Object*. [URI]
* *CybOXTM Version 2.1.1 Part 75: Win Kernel Object*. [URI]
* *CybOXTM Version 2.1.1 Part 76: Win Mailslot Object*. [URI]
* *CybOXTM Version 2.1.1 Part 77: Win Memory Page Region Object*. [URI]
* *CybOXTM Version 2.1.1 Part 78: Win Mutex Object*. [URI]
* *CybOXTM Version 2.1.1 Part 79: Win Network Route Entry Object*. [URI]
* *CybOXTM Version 2.1.1 Part 80: Win Network Share Object*. [URI]
* *CybOXTM Version 2.1.1 Part 81: Win Pipe Object*. [URI]
* *CybOXTM Version 2.1.1 Part 82: Win Prefetch Object*. [URI]
* *CybOXTM Version 2.1.1 Part 83: Win Process Object*. [URI]
* *CybOXTM Version 2.1.1 Part 84: Win Registry Key Object*. [URI]
* *CybOXTM Version 2.1.1 Part 85: Win Semaphore Object*. [URI]
* *CybOXTM Version 2.1.1 Part 86: Win Service Object*. [URI]
* *CybOXTM Version 2.1.1 Part 87: Win System Object*. [URI]
* *CybOXTM Version 2.1.1 Part 88: Win System Restore Object*. [URI]
* *CybOXTM Version 2.1.1 Part 89: Win Task Object*. [URI]
* *CybOXTM Version 2.1.1 Part 90: Win Thread Object*. [URI]
* *CybOXTM Version 2.1.1 Part 91: Win User Account Object*. [URI]
* *CybOXTM Version 2.1.1 Part 92: Win Volume Object*. [URI]
* *CybOXTM Version 2.1.1 Part 93: Win Waitable Timer Object*. [URI]
* *CybOXTM Version 2.1.1 Part 94: X509 Certificate Object*. [URI]

Related work:

This specification is related to:

* *STIXTM Version 1.2.1 (placeholder)*

Abstract:

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

Status:

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

URI patterns:

Initial publication URI:  
http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part1-overview/stix-v1.2.1-csd01-part1-overview.docx

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

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

Copyright © OASIS Open 2015. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full [Policy](https://www.oasis-open.org/policies-guidelines/ipr) may be found at the OASIS website.

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

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

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

Portions copyright © United States Government 2012-2015.  All Rights Reserved.  
  
STIX™, TAXII™, AND CybOX™ (STANDARD OR STANDARDS) AND THEIR COMPONENT PARTS ARE PROVIDED “AS IS” WITHOUT ANY WARRANTY OF ANY KIND, EITHER EXPRESSED, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTY THAT THESE STANDARDS OR ANY OF THEIR COMPONENT PARTS WILL CONFORM TO SPECIFICATIONS, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR FREEDOM FROM INFRINGEMENT, ANY WARRANTY THAT THE STANDARDS OR THEIR COMPONENT PARTS WILL BE ERROR FREE, OR ANY WARRANTY THAT THE DOCUMENTATION, IF PROVIDED, WILL CONFORM TO THE STANDARDS OR THEIR COMPONENT PARTS. IN NO EVENT SHALL THE UNITED STATES GOVERNMENT OR ITS CONTRACTORS OR SUBCONTRACTORS BE LIABLE FOR ANY DAMAGES, INCLUDING, BUT NOT LIMITED TO, DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES, ARISING OUT OF, RESULTING FROM, OR IN ANY WAY CONNECTED WITH THESE STANDARDS OR THEIR COMPONENT PARTS OR ANY PROVIDED DOCUMENTATION, WHETHER OR NOT BASED UPON WARRANTY, CONTRACT, TORT, OR OTHERWISE, WHETHER OR NOT INJURY WAS SUSTAINED BY PERSONS OR PROPERTY OR OTHERWISE, AND WHETHER OR NOT LOSS WAS SUSTAINED FROM, OR AROSE OUT OF THE RESULTS OF, OR USE OF, THE STANDARDS, THEIR COMPONENT PARTS, AND ANY PROVIDED DOCUMENTATION. THE UNITED STATES GOVERNMENT DISCLAIMS ALL WARRANTIES AND LIABILITIES REGARDING THE STANDARDS OR THEIR COMPONENT PARTS ATTRIBUTABLE TO ANY THIRD PARTY, IF PRESENT IN THE STANDARDS OR THEIR COMPONENT PARTS AND DISTRIBUTES IT OR THEM “AS IS.”

Table of Contents

[1 Introduction 7](#_Toc436998893)

[1.1 CybOXTM Specification Documents 7](#_Toc436998894)

[1.2 Document Conventions 7](#_Toc436998895)

[1.2.1 Fonts 7](#_Toc436998896)

[1.2.2 UML Package References 8](#_Toc436998897)

[1.2.3 UML Diagrams 8](#_Toc436998898)

[1.2.3.1 Class Properties 8](#_Toc436998899)

[1.2.3.2 Diagram Icons and Arrow Types 8](#_Toc436998900)

[1.2.4 Property Table Notation 9](#_Toc436998901)

[1.2.5 Property and Class Descriptions 9](#_Toc436998902)

[1.3 Terminology 10](#_Toc436998903)

[1.4 Normative References 10](#_Toc436998904)

[2 Background Information 11](#_Toc436998905)

[2.1 Cyber Observables 11](#_Toc436998906)

[2.2 Objects 11](#_Toc436998907)

[3 Data Model 12](#_Toc436998908)

[3.1 PDFFileObjectType Class 12](#_Toc436998909)

[3.2 PDFXRefTableListType Class 13](#_Toc436998910)

[3.3 PDFXRefTableType Class 14](#_Toc436998911)

[3.4 PDFXrefTableSubsectionListType Class 14](#_Toc436998912)

[3.5 PDFXrefTableSubsectionType Class 14](#_Toc436998913)

[3.6 PDFTrailerListType Class 15](#_Toc436998914)

[3.7 PDFTrailerType Class 15](#_Toc436998915)

[3.8 PDFFileIDType Class 17](#_Toc436998916)

[3.9 PDFIndirectObjectListType Class 17](#_Toc436998917)

[3.10 PDFIndirectObjectType Class 17](#_Toc436998918)

[3.11 PDFIndirectObjectIDType Class 18](#_Toc436998919)

[3.12 PDFIndirectObjectContentsType Class 18](#_Toc436998920)

[3.13 PDFStreamType Class 19](#_Toc436998921)

[3.14 PDFDocumentInformationDictionaryType Class 20](#_Toc436998922)

[3.15 PDFXrefEntryListType Class 21](#_Toc436998923)

[3.16 PDFXrefEntryType Class 21](#_Toc436998924)

[3.17 PDFDictionaryType Class 22](#_Toc436998925)

[3.18 PDFFileMetadataType Class 22](#_Toc436998926)

[3.19 PDFKeywordCountsType Class 23](#_Toc436998927)

[3.20 PDFKeywordCountType Class 25](#_Toc436998928)

[3.21 PDFObjectTypeEnum Enumeration 25](#_Toc436998929)

[3.22 PDFXrefEntryTypeEnum Enumeration 26](#_Toc436998930)

[4 Conformance 27](#_Toc436998931)

[Acknowledgments 28](#_Toc436998932)

[Revision History 29](#_Toc436998933)

# Introduction

[All text is normative unless otherwise labeled]

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

This document serves as the specification for the CybOX PDF File Object Version 2.1.1 data model, which is one of eighty-eight CybOX Object data models.

In Section **1.1** we discuss additional specification documents, in Section **1.2** we provide document conventions in, 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 PDF File Object data model. We present the PDF File Object data model specification details in Section **3** and conformance information in Section **4**.

## CybOXTM Specification Documents

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

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

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

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

## Document Conventions

The following conventions are used in this document.

### Fonts

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

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

Examples: Action, Object, Event, Property

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

Examples: ActionType, cyboxCommon:BaseObjectPropertyType

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

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

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

### UML Package References

Each CybOX data model is captured in a different UML package (e.g., Core package) where the packages together compose the full CybOX UML model. To refer to a particular class of a specific package, we use the format package\_prefix:class, where package\_prefix corresponds to the appropriate UML package. The [*CybOX Version 2.1.1 Part 1: Overview*](#AdditionalArtifacts) document contains the full list of CybOX packages, along with the associated prefix notations, descriptions, and examples.

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

### UML Diagrams

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

In UML diagrams, classes are often presented with their attributes elided, to avoid clutter. The fully described class can usually be found in a related diagram. A class presented with an empty section at the bottom of the icon indicates that there are no attributes other than those that are visualized using associations.

#### Class Properties

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

#### Diagram Icons and Arrow Types

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

Table ‑. UML diagram icons

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

### Property Table Notation

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

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

### Property and Class Descriptions

Each class and property defined in CybOX is described using the format, “The X property verbY.” For example, in the specification for the CybOX Core data model, we write, “The id property specifies a globally unique identifier for the Action.” In fact, the verb “specifies” could have been replaced by any number of alternatives: “defines,” “describes,” “contains,” “references,” etc.

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

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

|  |  |
| --- | --- |
| **Verb** | **CybOX Definition** |
| captures | Used to record and preserve information without implying anything about the structure of a class or property. Often used for properties that encompass general content. This is the least precise of the three verbs. |
|  | *Examples*:  The Observable\_Source property characterizes the source of the Observable information. Examples of details captured include identifying characteristics, time-related attributes, and a list of the tools used to collect the information.  The Description property captures a textual description of the Action. |
| characterizes | Describes the distinctive nature or features of a class or property. Often used to describe classes and properties that themselves comprise one or more other properties. |
|  | *Examples*:  The Action property characterizes a cyber observable Action.  The Obfuscation\_Technique property characterizes a technique an attacker could potentially leverage to obfuscate the Observable. |
| specifies | Used to clearly and precisely identify particular instances or values associated with a property. Often used for properties that are defined by a controlled vocabulary or enumeration; typically used for properties that take on only a single value. |
|  | *Example*:  The cybox\_major\_version property specifies the major version of the CybOX language used for the set of Observables. |

## Terminology

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

## Normative References

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

# Background Information

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

## Cyber Observables

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

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

## Objects

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

# Data Model

## PDFFileObjectType Class

The PDFFileObjectType class is intended to characterize the structural makeup of PDF files. The UML diagram corresponding to the PDFFileObjectType class is shown in **Figure 3‑1**.

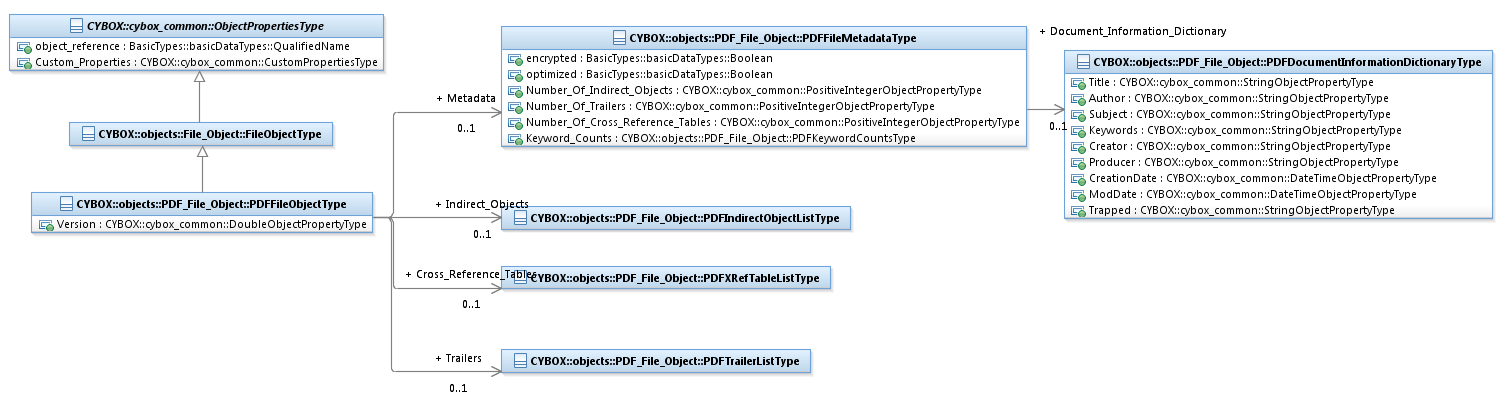


Figure ‑. UML diagram of the PDFFileObjectType class

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

Table ‑. Properties of the PDFFileObjectType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **Metadata** | PDFFileObj:  PDFFileMetadataType | 0..1 | The Metadata property captures some useful metadata associated with the PDF file. |
| **Version** | cyboxCommon:  DoubleObjectPropertyType | 0..1 | The Version property specifies the decimal version number portion of the string from the PDF Header that specifies the version of the PDF specification to which the PDF file conforms, e.g. '1.4'. |
| **Indirect\_Objects** | PDFFileObj:  PDFIndirectObjectListType | 0..1 | The Indirect\_Objects property captures the indirect objects included in the PDF file, representing the contents of a document. |
| **Cross\_Reference\_Tables** | PDFFileObj:  PDFXRefTableListType | 0..1 | The Cross\_Reference\_Tables property captures the cross-reference tables included in the PDF file, used for facilitating random access of indirect PDF objects. |
| **Trailers** | PDFFileObj:  PDFTrailerListType | 0..1 | The Trailers property captures the trailers included in the PDF file, used for capturing offsets to the cross-reference table and important objects. |

## PDFXRefTableListType Class

The PDFXRefTableListType class captures a list of PDF cross-reference tables. The UML diagram corresponding to the PDFXRefTableListType class is shown in Figure 3‑2.

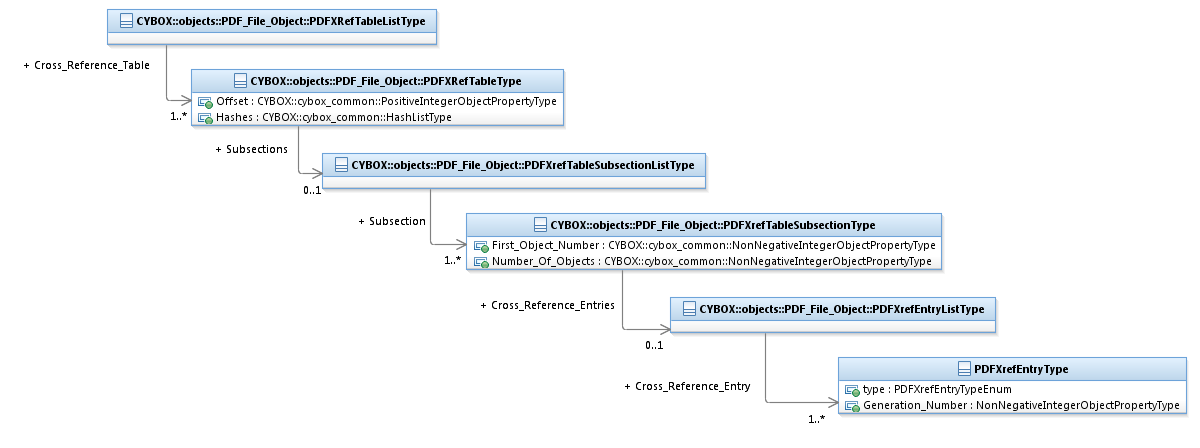


Figure ‑. UML diagram for the PDFXRefTableListType class

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

Table ‑. Properties of the PDFXRefTableListType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **Cross\_Reference\_Table** | PDFFileObj:  PDFXRefTableType | 1..\* | The Cross\_Reference\_Table property captures the cross-reference table contained in the PDF file, for the random access of indirect objects contained in the file. |

## PDFXRefTableType Class

The PDFXRefTableType class captures the details of a PDF cross-reference table, which provides a capability for the random access of indirect objects contained in the file.

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

Table ‑. Properties of the PDFXRefTableType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **Subsections** | PDFFileObj:  PDFXrefTableSubsectionListType | 0..1 | The Subsections property captures the subsections contained in the cross-reference table. |
| **Offset** | cyboxCommon:  PositiveIntegerObjectPropertyType | 0..1 | The Offset property specifies the offset of the cross-reference from the beginning of the file, in bytes. |
| **Hashes** | cyboxCommon:HashListType | 0..1 | The Hashes property captures any hashes that were computed for the cross-reference table. |

## PDFXrefTableSubsectionListType Class

The PDFXrefTableSubsectionListType class captures a list of cross-reference table subsections.

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

Table ‑. Properties of the PDFXrefTableSubsectionListType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **Subsection** | PDFFileObj:  PDFXrefTableSubsectionType | 1..\* | The Subsection property captures a single cross-reference table subsection in the list. |

## PDFXrefTableSubsectionType Class

The PDFXrefTableSubsectionType class captures details of subsections contained within a PDF cross-reference table.

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

Table ‑. Properties of the PDFXrefTableSubsectionType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **First\_Object\_Number** | cyboxCommon:  NonNegativeIntegerObjectPropertyType | 0..1 | The First\_Object\_Number property captures the object number of the first object for which there is a corresponding entry in this cross-reference subsection. |
| **Number\_Of\_Objects** | cyboxCommon:  NonNegativeIntegerObjectPropertyType | 0..1 | The Number\_Of\_Objects property captures the number of objects for which there are corresponding entries in this cross-reference subsection. |
| **Cross\_Reference\_Entries** | PDFFileObj:  PDFXrefEntryListType | 0..1 | The Cross\_Reference\_Entries property specifies the cross-reference entries contained in this cross-reference subsection. |

## PDFTrailerListType Class

The PDFTrailerListType class captures a list of PDF trailers. The UML diagram corresponding to the PDFTrailerListType class is shown in Figure 3‑3.

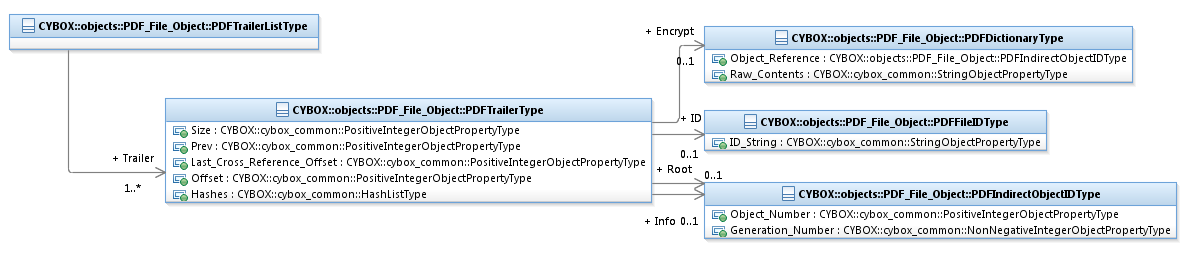


Figure ‑. UML diagram for the PDFTrailerListType class

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

Table ‑. Properties of the PDFTrailerListType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **Trailer** | PDFFileObj:  PDFTrailerType | 1..\* | The Trailer property captures a PDF file trailer contained in the file, used by applications for quickly locating the cross-reference table and certain special objects. |

## PDFTrailerType Class

The PDFTrailerType class captures the details of a PDF trailer.

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

Table ‑. Properties of the PDFTrailerType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **Size** | cyboxCommon:  PositiveIntegerObjectPropertyType | 0..1 | The Size property captures the total number of entries in the file's cross-reference table. |
| **Prev** | cyboxCommon:  PositiveIntegerObjectPropertyType | 0..1 | The Prev property the byte offset from the beginning of the file to the beginning of the previous cross-reference table. This is only applicable for files that have more than one cross-reference table. |
| **Root** | PDFFileObj:  PDFIndirectObjectIDType | 0..1 | The Root property captures an indirect object reference that points to the catalog dictionary for the PDF document contained in the file. |
| **Encrypt** | PDFFileObj:PDFDictionaryType | 0..1 | The Encrypt property captures the PDF document's encryption dictionary, either through an indirect reference or embedded set of key/value pairs. |
| **Info** | PDFFileObj:  PDFIndirectObjectIDType | 0..1 | The Info property captures an indirect object reference that points to the document information dictionary. |
| **ID** | PDFFileObj:PDFFileIDType | 0..1 | The ID property captures an array of two strings that constitutes a file identifier. |
| **Last\_Cross\_**  **Reference\_Offset** | cyboxCommon:  PositiveIntegerObjectPropertyType | 0..1 | The Last\_Cross\_Reference\_Offset property captures the byte offset, relative to the beginning of the file, of the last cross-reference table contained in the file. |
| **Offset** | cyboxCommon:  PositiveIntegerObjectPropertyType | 0..1 | The Offset property specifies the offset of the trailer from the beginning of the file, in bytes. |
| **Hashes** | cyboxCommon:HashListType | 0..1 | The Hashes property captures any hashes that were computed for the trailer. |

## PDFFileIDType Class

The PDFFileIDType class captures the details of a PDF ID value stored in a trailer.

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

Table ‑. Properties of the PDFFileIDType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **ID\_String** | cyboxCommon:  StringObjectPropertyType | 1..2 | The ID\_String property captures one of the two strings that constitutes the file identifier. |

## PDFIndirectObjectListType Class

The PDFIndirectObjectListType class captures a list of PDF indirect objects.

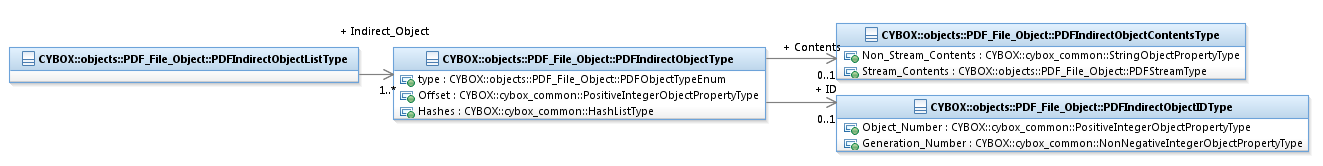


Figure ‑. UML diagram of the PDFIndirectObjectListType class

The property table of the PDFIndirectObjectListType class is given in **Table 3‑9**.

Table ‑. Properties of the PDFIndirectObjectListType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **Indirect\_Object** | PDFFileObj:  PDFIndirectObjectType | 1..\* | The Indirect\_Object property captures a single PDF indirect object contained in the file. |

## PDFIndirectObjectType Class

The PDFIndirectObjectType class captures the details of a PDF document indirect object, used in constructing and storing data associated with the PDF document.

The property table of the PDFIndirectObjectType class is given in **Table 3‑10**.

Table ‑. Properties of the PDFIndirectObjectType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **type** | PDFFileObj:PDFObjectTypeEnum | 0..1 | The type property specifies the basic type of the PDF indirect object. |
| **ID** | PDFFileObj:  PDFIndirectObjectIDType | 0..1 | The ID property specifies the identifier of the PDF indirect object, consisting of an object number and generation number. |
| **Contents** | PDFFileObj:  PDFIndirectObjectContentsType | 0..1 | The Contents property captures the contents of the PDF indirect object, including non-stream and stream data. |
| **Offset** | cyboxCommon:  PositiveIntegerObjectPropertyType | 0..1 | The Offset property specifies the offset of the PDF indirect object from the beginning of the file, in bytes. |
| **Hashes** | cyboxCommon:HashListType | 0..1 | The Hashes property captures any hashes that were computed for the PDF indirect object. |

## PDFIndirectObjectIDType Class

The PDFIndirectObjectIDType class captures the details of PDF indirect object IDs.

The property table of the PDFIndirectObjectIDType class is given in **Table 3‑11**.

Table ‑. Properties of the PDFIndirectObjectIDType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **Object\_Number** | cyboxCommon:  PositiveIntegerObjectPropertyType | 0..1 | The Object\_Number property captures the number portion of the indirect object ID. |
| **Generation\_Number** | cyboxCommon:  NonNegativeIntegerObjectPropertyType | 0..1 | The Generation\_Number property captures the generation number portion of the indirect object ID. |

## PDFIndirectObjectContentsType Class

The PDFIndirectObjectContentsType class captures the contents of a PDF indirect object, including both stream and non-stream portions.

The property table of the PDFIndirectObjectContentsType class is given in **Table 3‑12**.

Table ‑. Properties of the PDFIndirectObjectContentsType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **Non\_Stream\_Contents** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Non\_Stream\_Contents property captures the raw contents of the PDF indirect object excluding any stream data (i.e. everything after the 'obj' keyword and before the 'endobj' keyword up to but not including anything between the 'stream' and 'endstream' keywords) as a string enclosed in an XML CDATA section. |
| **Stream\_Contents** | PDFFileObj:PDFStreamType | 0..1 | The Stream\_Contents property captures the stream contained within in the PDF indirect object, if applicable. |

## PDFStreamType Class

The PDFStreamType class element captures details of PDF document stream objects, which represent arbitrary sequences of bytes.

The property table of the PDFStreamType class is given in **Table 3‑13**.

Table ‑. Properties of the PDFStreamType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **Raw\_Stream** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Raw\_Stream property captures the raw, undecoded stream (i.e., everything between the 'stream' and 'endstream' keywords), as a hex string. |
| **Raw\_Stream\_Hashes** | cyboxCommon:HashListType | 0..1 | The Raw\_Stream\_Hashes property captures any hashes of the raw, undecoded stream. |
| **Decoded\_Stream** | cyboxCommon:  HexBinaryObjectPropertyType | 0..1 | The Decoded\_Stream property captures the decoded stream (i.e., after undoing the specified filters in the correct order) as a hex string. |
| **Decoded\_Stream\_Hashes** | cyboxCommon:HashListType | 0..1 | The Decoded\_Stream\_Hashes property captures any hashes of the decoded stream. |

## PDFDocumentInformationDictionaryType Class

The PDFDocumentInformationDictionaryType class captures details of the PDF Document Information Dictionary, used for storing metadata associated with the PDF document.

The property table of the PDFDocumentInformationDictionaryType class is given in **Table 3‑14**.

Table ‑. Properties of the PDFDocumentInformationDictionaryType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **Title** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Title property captures the title for the PDFDocumentInformationDictionary and reflects what the content producer thinks the PDFDocumentInformationDictionary as a whole should be called. The Title property is typically used by humans to reference a particular PDFDocumentInformationDictionary; however, it is not suggested for correlation. |
| **Author** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Author property captures the name of the person who created the PDF document. |
| **Subject** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Subject property captures the subject of the PDF document. |
| **Keywords** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Keywords property captures the keywords associated with the PDF document. |
| **Creator** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Creator property captures the name of the application that created the original document, for cases where the original document was then converted to PDF. |
| **Producer** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Producer property captures the name of the application that converted the document to PDF, for cases where the original document was then converted to PDF. |
| **CreationDate** | cyboxCommon:  DateTimeObjectPropertyType | 0..1 | The CreationDate property captures the date and time that the document was created. |
| **ModDate** | cyboxCommon:  DateTimeObjectPropertyType | 0..1 | The ModDate property captures the date and time that the document was most recently modified. |
| **Trapped** | cyboxCommon:  StringObjectPropertyType | 0..1 | The Trapped property captures a name object indicating whether the document has been modified to include trapping information. |

## PDFXrefEntryListType Class

The PDFXrefEntryListType class captures a list of cross-reference table subsection entries.

The property table of the PDFXrefEntryListType class is given in **Table 3‑15**.

Table ‑. Properties of the PDFXrefEntryListType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **Cross\_Reference\_Entry** | PDFFileObj:  PDFXrefEntryType | 1..\* | The Cross\_Reference\_Entry property captures a single cross-reference subsection entry in the list. |

## PDFXrefEntryType Class

The PDFXrefEntryType class captures details of a cross-reference table subsection entry.

The property table of the PDFXrefEntryType class is given in **Table 3‑16**.

Table ‑. Properties of the PDFXrefEntryType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **type** | PDFFileObj:PDFXrefEntryTypeEnum | 0..1 | The type property specifies the type of the cross-reference entry. |
| **Byte\_Offset** | cyboxCommon:  IntegerObjectPropertyType | 1 | The Byte\_Offset property captures the 10-digit number, padded with leading zeros if necessary, that specifies the number of bytes from the beginning of the file to the beginning of the object. |
| **Object\_Number** | cyboxCommon:  NonNegativeIntegerObjectPropertyType | 1 | The Object\_Number property specifies the 10-digit object number of the next free object. |
| **Generation\_Number** | cyboxCommon:  NonNegativeIntegerObjectPropertyType | 0..1 | The Generation\_Number property specifies the 5-digit generation number to be used when an object with the same object number is created. |

## PDFDictionaryType Class

The PDFDictionaryType class captures a PDF dictionary as a set of key value pairs, or as a reference to an indirect object that contains.

The property table of the PDFDictionaryType class is given in **Table 3‑17**.

Table ‑. Properties of the PDFDictionaryType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **Object\_Reference** | PDFFileObj:  PDFIndirectObjectIDType | 1 | The Object\_Reference property captures a reference to an indirect PDF object that contains the dictionary, via its object and generation numbers. |
| **Raw\_Contents** | cyboxCommon:  StringObjectPropertyType | 1 | The Raw\_Contents property captures the contents of the dictionary as a string enclosed in an XML CDATA section. |

## PDFFileMetadataType Class

The PDFFileMetadataType class captures some metadata regarding the PDF file object.

The property table of the PDFFileMetadataType class is given in **Table 3‑18**.

Table ‑. Properties of the PDFFileMetadataType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **encrypted** | basicDataTypes:Boolean | 0..1 | The encrypted property specifies whether the PDF file is encrypted. |
| **optimized** | basicDataTypes:Boolean | 0..1 | The optimized property specifies whether the PDF file has been optimized. |
| **Document\_Information\_Dictionary** | PDFFileObj:  PDFDocumentInformationDictionaryType | 0..1 | The Document\_Information\_Dictionary property captures the details of the PDF Document Information Dictionary, which includes properties like the document creation date and producer, if present in the PDF document. |
| **Number\_Of\_Indirect\_Objects** | cyboxCommon:  PositiveIntegerObjectPropertyType | 0..1 | The Number\_Of\_Indirect\_Objects property captures the number of indirect PDF objects contained in the file. |
| **Number\_Of\_Trailers** | cyboxCommon:  PositiveIntegerObjectPropertyType | 0..1 | The Number\_Of\_Trailers property captures the number of trailers contained in the file. |
| **Number\_Of\_Cross\_Reference\_Tables** | cyboxCommon:  PositiveIntegerObjectPropertyType | 0..1 | The Number\_Of\_Cross\_Reference\_Tables property captures the number of cross-reference tables contained in the file. |
| **Keyword\_Counts** | PDFFileObj:  PDFKeywordCountsType | 0..1 | The Keyword\_Counts property captures the counts of various PDF keyword names in the file. |

## PDFKeywordCountsType Class

The PDFKeywordCountsType class captures the occurrences of various keywords in a PDF file.

The property table of the PDFKeywordCountsType class is given in **Table 3‑19**.

Table ‑. Properties of the PDFKeywordCountsType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **Page\_Count** | PDFFileObj:  PDFKeywordCountType | 0..1 | The Page\_Count property captures the number of occurrences of the '/Page' keyword in the PDF file, which provides an indication of the number of pages in the PDF document. |
| **Encrypt\_Count** | PDFFileObj:  PDFKeywordCountType | 0..1 | The Encrypt\_Count property captures the number of occurrences of the '/Encrypt' keyword in the PDF file, which indicates that the PDF uses encryption. |
| **ObjStm\_Count** | PDFFileObj:  PDFKeywordCountType | 0..1 | The ObjStm\_Count property captures the number of occurrences of the '/ObjStm' keyword in the PDF file. |
| **JS\_Count** | PDFFileObj:  PDFKeywordCountType | 0..1 | The JS\_Count property captures the number of occurrences of the '/JS' keyword in the PDF file. |
| **JavaScript\_Count** | PDFFileObj:  PDFKeywordCountType | 0..1 | The JavaScript\_Count property captures the number of occurrences of the '/JavaScript' keyword in the PDF file. |
| **AA\_Count** | PDFFileObj:  PDFKeywordCountType | 0..1 | The AA\_Count property captures the number of occurrences of the '/AA' keyword in the PDF file. |
| **OpenAction\_Count** | PDFFileObj:  PDFKeywordCountType | 0..1 | The OpenAction\_Count property captures the number of occurrences of the '/OpenAction' keyword in the PDF file. |
| **ASCIIHexDecode\_Count** | PDFFileObj:  PDFKeywordCountType | 0..1 | The ASCIIHexDecode\_Count property captures the number of occurrences of the '/ASCIIHexDecode' keyword in the PDF file. |
| **ASCII85Decode\_Count** | PDFFileObj:  PDFKeywordCountType | 0..1 | The ASCII85Decode\_Count property captures the number of occurrences of the '/ASCII85Decode' keyword in the PDF file. |
| **LZWDecode\_Count** | PDFFileObj:  PDFKeywordCountType | 0..1 | The LZWDecode\_Count property captures the number of occurrences of the '/LZWDecode' keyword in the PDF file. |
| **FlateDecode\_Count** | PDFFileObj:  PDFKeywordCountType | 0..1 | The FlateDecode\_Count property captures the number of occurrences of the '/FlateDecode' keyword in the PDF file. |
| **RunLengthDecode\_Count** | PDFFileObj:  PDFKeywordCountType | 0..1 | The RunLengthDecode\_Count property captures the number of occurrences of the '/RunLengthDecode' keyword in the PDF file. |
| **JBIG2Decode\_Count** | PDFFileObj:  PDFKeywordCountType | 0..1 | The JBIG2Decode\_Count property captures the number of occurrences of the '/JBIG2Decode' keyword in the PDF file. |
| **DCTDecode\_Count** | PDFFileObj:  PDFKeywordCountType | 0..1 | The DCTDecode\_Count property captures the number of occurrences of the '/DCTDecode' keyword in the PDF file. |
| **RichMedia\_Count** | PDFFileObj:  PDFKeywordCountType | 0..1 | The RichMedia\_Count property captures the number of occurrences of the '/RichMedia' keyword in the PDF file. |
| **CCITTFaxDecode\_Count** | PDFFileObj:  PDFKeywordCountType | 0..1 | The CCITTFaxDecode\_Count property captures the number of occurrences of the '/CCITTFaxDecode' keyword in the PDF file. |
| **Launch\_Count** | PDFFileObj:  PDFKeywordCountType | 0..1 | The Launch\_Count property captures the number of occurrences of the '/Launch' keyword in the PDF file. |
| **XFA\_Count** | PDFFileObj:  PDFKeywordCountType | 0..1 | The XFA\_Count property captures the number of occurrences of the '/XFA' keyword in the PDF file. |

## PDFKeywordCountType Class

The PDFKeywordCountType class captures the obfuscated and non-obfuscated occurrences of a keyword.

The property table of the PDFKeywordCountType class is given in **Table 3‑20**.

Table ‑. Properties of the PDFKeywordCountType class

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Multiplicity** | **Description** |
| **Non\_Obfuscated\_Count** | cyboxCommon:  NonNegativeIntegerObjectPropertyType | 0..1 | The Non\_Obfuscated\_Count property captures the number of times the keyword occurred in the PDF file without any obfuscation. |
| **Obfuscated\_Count** | cyboxCommon:  NonNegativeIntegerObjectPropertyType | 0..1 | The Obfuscated\_Count property captures the number of times the keyword occurred in the PDF file with some form of obfuscation, such as with hexcodes. |

## PDFObjectTypeEnum Enumeration

The literals of the PDFObjectTypeEnum enumeration are given in **Table 3‑21**.

Table ‑. Literals of the PDFObjectTypeEnum enumeration

|  |  |
| --- | --- |
| **Enumeration Literal** | **Description** |
| **Boolean** |  |
| **Integer** |  |
| **String** |  |
| **Name** |  |
| **Array** |  |
| **Dictionary** |  |
| **Stream** |  |
| **Null** |  |

## PDFXrefEntryTypeEnum Enumeration

The literals of the PDFXrefEntryTypeEnum enumeration are given in **Table 3‑22**.

Table ‑. Literals of the PDFXrefEntryTypeEnum enumeration

|  |  |
| --- | --- |
| **Enumeration Literal** | **Description** |
| **In-Use** |  |
| **Free** |  |

# Conformance

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

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

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

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

Acknowledgments

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

Participants:

Dean Thompson, Australia and New Zealand Banking Group (ANZ Bank)

Bret Jordan, Blue Coat Systems, Inc.

Adnan Baykal, Center for Internet Security (CIS)

Liron Schiff, Comilion (mobile) Ltd.

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

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

Ryusuke Masuoka, Fujitsu Limited

Eric Burger, Georgetown University

Jason Keirstead, IBM

Paul Martini, iboss, Inc.

Jerome Athias, Individual

Sanjiv Kalkar, Individual

Terry MacDonald, Individual

Alex Pinto, Individual

Patrick Maroney, Integrated Networking Technologies, Inc.

Wouter Bolsterlee, Intelworks BV

Joep Gommers, Intelworks BV

Sergey Polzunov, Intelworks BV

Rutger Prins, Intelworks BV

Andrei Sîrghi, Intelworks BV

Jonathan Baker, MITRE Corporation

Sean Barnum, MITRE Corporation

Mark Davidson, MITRE Corporation

Ivan Kirillov, MITRE Corporation

John Wunder, MITRE Corporation

Mike Boyle, National Security Agency

Jessica Fitzgerald-McKay, National Security Agency

Takahiro Kakumaru, NEC Corporation

John-Mark Gurney, New Context Services, Inc.

Christian Hunt, New Context Services, Inc.

Andrew Storms, New Context Services, Inc.

Igor Baikalov, Securonix

Bernd Grobauer, Siemens AG

John Anderson, Soltra

Trey Darley, Soltra

Paul Dion, Soltra

Brandon Hanes, Soltra

Ali Khan, Soltra

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

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Editor** | **Changes Made** |
| wd01 | 16 November 2015 | Desiree Beck Trey Darley Ivan Kirillov Rich Piazza | Initial transfer to OASIS template |