



CybOX™ Version 2.1.1. Part 43: PDF File Object

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/part43-pdf-file/cybox-v2.1.1-csprd01-part43-pdf-file.docx> (Authoritative)
<http://docs.oasis-open.org/cti/cybox/v2.1.1/csprd01/part43-pdf-file/cybox-v2.1.1-csprd01-part43-pdf-file.html>
<http://docs.oasis-open.org/cti/cybox/v2.1.1/csprd01/part43-pdf-file/cybox-v2.1.1-csprd01-part43-pdf-file.pdf>

Previous version:

N/A

Latest version:

<http://docs.oasis-open.org/cti/cybox/v2.1.1/part43-pdf-file/cybox-v2.1.1-part43-pdf-file.docx>
(Authoritative)
<http://docs.oasis-open.org/cti/cybox/v2.1.1/part43-pdf-file/cybox-v2.1.1-part43-pdf-file.html>
<http://docs.oasis-open.org/cti/cybox/v2.1.1/part43-pdf-file/cybox-v2.1.1-part43-pdf-file.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 PDF File Object data model, which is one of the Object data models for CybOX content.

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-pdf-file]

CybOX™ Version 2.1.1. Part 43: PDF File Object. 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/part43-pdf-file/cybox-v2.1.1-csprd01-part43-pdf-file.html>. Latest version: <http://docs.oasis-open.org/cti/cybox/v2.1.1/part43-pdf-file/cybox-v2.1.1-part43-pdf-file.html>.

Notices

Copyright © OASIS Open 2016. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full [Policy](#) 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.

Portions copyright © United States Government 2012-2016. All Rights Reserved.

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

WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR FREEDOM FROM INFRINGEMENT, ANY WARRANTY THAT THE STANDARDS OR THEIR COMPONENT PARTS WILL BE ERROR FREE, OR ANY WARRANTY THAT THE DOCUMENTATION, IF PROVIDED, WILL CONFORM TO THE STANDARDS OR THEIR COMPONENT PARTS. IN NO EVENT SHALL THE UNITED STATES GOVERNMENT OR ITS CONTRACTORS OR SUBCONTRACTORS BE LIABLE FOR ANY DAMAGES, INCLUDING, BUT NOT LIMITED TO, DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES, ARISING OUT OF, RESULTING FROM, OR IN ANY WAY CONNECTED WITH THESE STANDARDS OR THEIR COMPONENT PARTS OR ANY PROVIDED DOCUMENTATION, WHETHER OR NOT BASED UPON WARRANTY, CONTRACT, TORT, OR OTHERWISE, WHETHER OR NOT INJURY WAS SUSTAINED BY PERSONS OR PROPERTY OR OTHERWISE, AND WHETHER OR NOT LOSS WAS SUSTAINED FROM, OR AROSE OUT OF THE RESULTS OF, OR USE OF, THE STANDARDS, THEIR COMPONENT PARTS, AND ANY PROVIDED DOCUMENTATION. THE UNITED STATES GOVERNMENT DISCLAIMS ALL WARRANTIES AND LIABILITIES REGARDING THE STANDARDS OR THEIR COMPONENT PARTS ATTRIBUTABLE TO ANY THIRD PARTY, IF PRESENT IN THE STANDARDS OR THEIR COMPONENT PARTS AND DISTRIBUTES IT OR THEM "AS IS."

Table of Contents

1	Introduction.....	6
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	Property Table Notation.....	8
1.2.5	Property and Class Descriptions.....	8
1.3	Terminology.....	9
1.4	Normative References.....	9
2	Background Information.....	10
2.1	Cyber Observables.....	10
2.2	Objects.....	10
3	Data Model.....	11
3.1	PDFFileObjectType Class.....	11
3.2	PDFXRefTableListType Class.....	12
3.3	PDFXRefTableType Class.....	14
3.4	PDFXrefTableSubsectionListType Class.....	14
3.5	PDFXrefTableSubsectionType Class.....	15
3.6	PDFTrailerListType Class.....	15
3.7	PDFTrailerType Class.....	16
3.8	PDFFileIDType Class.....	18
3.9	PDFIndirectObjectListType Class.....	18
3.10	PDFIndirectObjectType Class.....	19
3.11	PDFIndirectObjectIDType Class.....	19
3.12	PDFIndirectObjectContentsType Class.....	20
3.13	PDFStreamType Class.....	21
3.14	PDFDocumentInformationDictionaryType Class.....	21
3.15	PDFXrefEntryListType Class.....	23
3.16	PDFXrefEntryType Class.....	23
3.17	PDFDictionaryType Class.....	25
3.18	PDFFileMetadataType Class.....	26
3.19	PDFKeywordCountsType Class.....	27
3.20	PDFKeywordCountType Class.....	29
3.21	PDFObjectTypeEnum Enumeration.....	30
3.22	PDFXrefEntryTypeEnum Enumeration.....	31
4	Conformance.....	32
	Appendix A. Acknowledgments.....	33
	Appendix B. Revision History.....	37

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 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, 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.

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.

The `package_prefix` for the PDF File data model is `PDFFileObj`. 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.

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.

Certain UML classes are associated with the UML stereotype `<<choice>>`. The `<<choice>>` stereotype specifies that only one of the available properties of the class can be populated at any time. The CybOX UML models utilize `Has_Choice` as the role/property name for associations to `<<choice>>` stereotyped classes. This property is a modeling convention rather than a native element of the underlying data model and acts as a placeholder for one of the available properties of the `<<choice>>` stereotyped class.

1.2.3.1 Class Properties

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

1.2.3.2 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 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.

1.2.5 Property and Class Descriptions

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

However, we thought that using a wide variety of verb phrases might confuse a reader of a specification document because the meaning of each verb could be interpreted slightly differently. On the other hand,

we didn't want to use a single, generic verb, such as "describes," because although the different verb choices may or may not be meaningful from an implementation standpoint, a distinction could be useful to those interested in the modeling aspect of CybOX.

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

Verb	CybOX Definition
<u>captures</u>	Used to record and preserve information without implying anything about the structure of a class or property. Often used for properties that encompass general content. This is the least precise of the three verbs.
	<p><i>Examples:</i></p> <p>The <code>Observable_Source</code> property characterizes the source of the Observable information. Examples of details <u>captured</u> include identifying characteristics, time-related attributes, and a list of the tools used to collect the information.</p> <p>The <code>Description</code> property <u>captures</u> a textual description of the Action.</p>
<u>characterizes</u>	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.
	<p><i>Examples:</i></p> <p>The <code>Action</code> property <u>characterizes</u> a cyber observable Action.</p> <p>The <code>Obfuscation_Technique</code> property <u>characterizes</u> a technique an attacker could potentially leverage to obfuscate the Observable.</p>
<u>specifies</u>	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.
	<p><i>Example:</i></p> <p>The <code>cybox_major_version</code> property <u>specifies</u> the major version of the CybOX language used for the set of Observables.</p>

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 PDF File Object data model that is necessary to fully understand the specification details given in Section 3.

2.1 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.

2.2 Objects

Cyber observable objects (Files, IP Addresses, etc) in CybOX are characterized with a combination of two levels of data models.

The first level is the Object data model which specifies a base set of properties universal to all types of Objects and enables them to integrate with the overall cyber observable framework specified in the CybOX Core data model.

The second level are the object property models which specify the properties of a particular type of Object via individual data models each focused on 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.

Any specific instance of an Object is represented utilizing the particular object properties data model within the general Object data model.

3 Data Model

3.1 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](#).

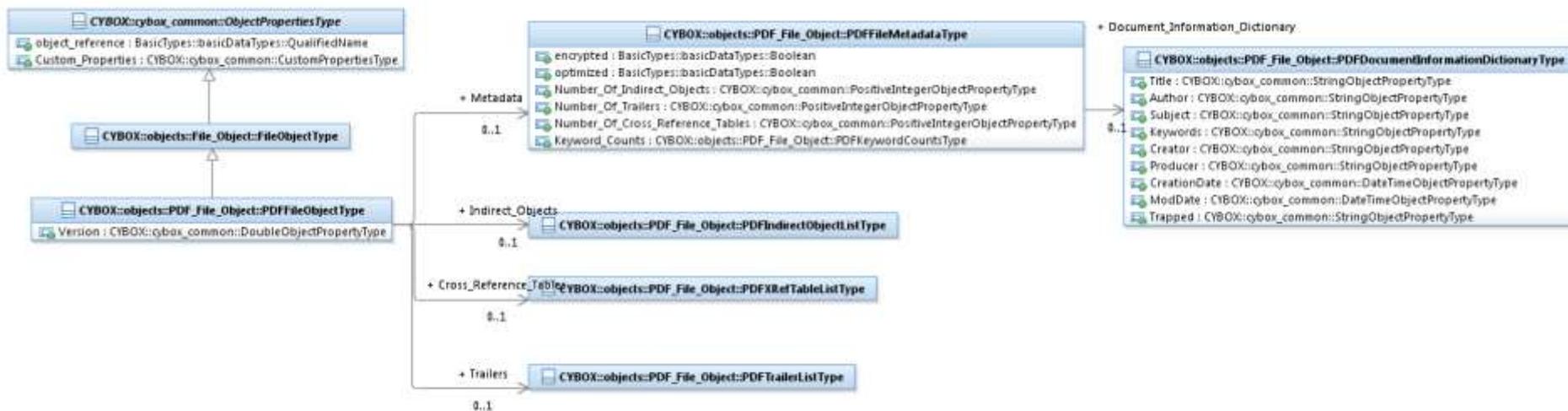


Figure 3-1. UML diagram of the PDFFileObjectType class

The property table of the PDFFileObjectType class is given in [Table 3-1](#).

Table 3-1. Properties of the PDFFileObjectType class

Name	Type	Multiplicity	Description
Metadata	PDFFileMetadataType	0..1	The Metadata property captures some useful metadata

			associated with the PDF file.
Version	cyboxCommon: DoubleObjectPropertyType	0..1	The <code>Version</code> 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	PDFIndirectObjectListType	0..1	The <code>Indirect_Objects</code> property captures the indirect objects included in the PDF file, representing the contents of a document.
Cross_Reference_Tables	PDFXRefTableListType	0..1	The <code>Cross_Reference_Tables</code> property captures the cross-reference tables included in the PDF file, used for facilitating random access of indirect PDF objects.
Trailers	PDFTrailerListType	0..1	The <code>Trailers</code> property captures the trailers included in the PDF file, used for capturing offsets to the cross-reference table and important objects.

3.2 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](#).

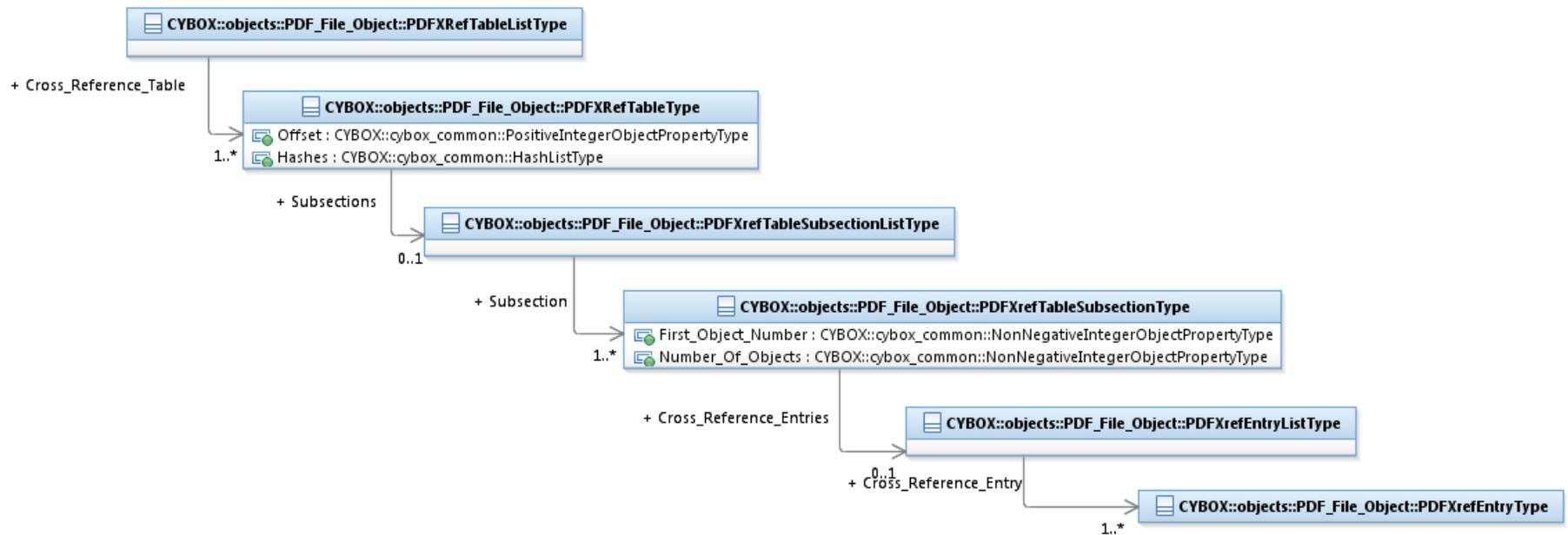


Figure 3-2. UML diagram for the PDFXRefTableListType class

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

Table 3-2. Properties of the PDFXRefTableListType class

Name	Type	Multiplicity	Description
Cross_Reference_Table	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.

3.3 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 3-3. Properties of the `PDFXRefTableType` class

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

3.4 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 3-4. Properties of the `PDFXrefTableSubsectionListType` class

Name	Type	Multiplicity	Description
Subsection	<code>PDFXrefTableSubsectionType</code>	1..*	The <code>Subsection</code> property captures a single cross-reference table subsection in the list.

3.5 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 3-5. Properties of the `PDFXrefTableSubsectionType` class

Name	Type	Multiplicity	Description
First_Object_Number	<code>cyboxCommon:</code> <code>NonNegativeIntegerObjectPropertyType</code>	0..1	The <code>First_Object_Number</code> property captures the object number of the first object for which there is a corresponding entry in this cross-reference subsection.
Number_Of_Objects	<code>cyboxCommon:</code> <code>NonNegativeIntegerObjectPropertyType</code>	0..1	The <code>Number_Of_Objects</code> property captures the number of objects for which there are corresponding entries in this cross-reference subsection.
Cross_Reference_Entries	<code>PDFXrefEntryListType</code>	0..1	The <code>Cross_Reference_Entries</code> property specifies the cross-reference entries contained in this cross-reference subsection.

3.6 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](#).

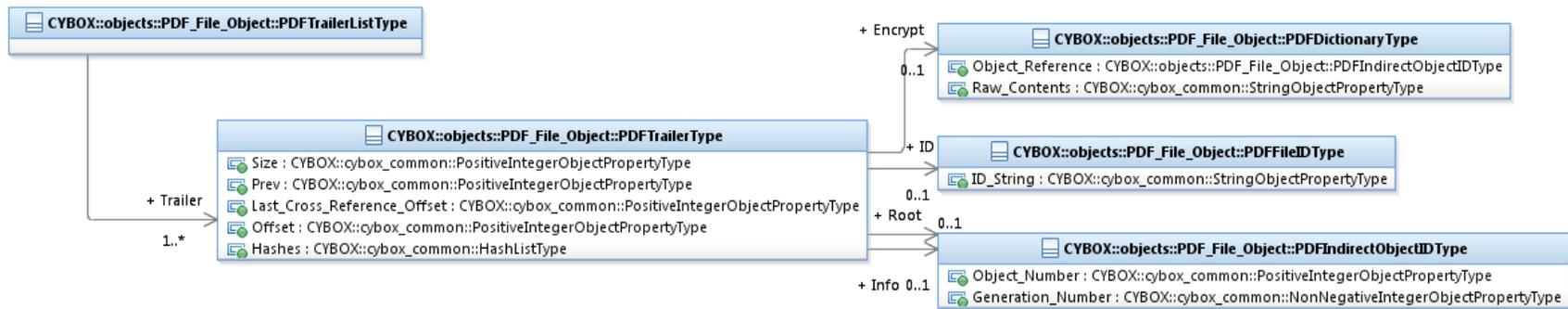


Figure 3-3. UML diagram for the PDFTrailerListType class

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

Table 3-6. Properties of the PDFTrailerListType class

Name	Type	Multiplicity	Description
Trailer	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.

3.7 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 3-7. Properties of the PDFTrailerType class

Name	Type	Multiplicity	Description
Size	cyboxCommon:	0..1	The Size property captures the total number of entries in the file's cross-reference table.

	PositiveIntegerObjectPropertyType		
Prev	cyboxCommon: PositiveIntegerObjectPropertyType	0..1	The Prev property captures 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	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	PDFDictionaryType	0..1	The Encrypt property captures the PDF document's encryption dictionary through either an indirect reference or an embedded set of key/value pairs.
Info	PDFIndirectObjectIDType	0..1	The Info property captures an indirect object reference that points to the document information dictionary.
ID	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.

3.8 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 3-8. 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.

3.9 PDFIndirectObjectListType Class

The PDFIndirectObjectListType class captures a list of PDF indirect objects. The UML diagram corresponding to the PDFIndirectObjectListType class is shown in [Figure 3-4](#).



Figure 3-4. UML diagram of the PDFIndirectObjectListType class

The property table of the PDFIndirectObjectListType class is given in [Table 3-9](#).

Table 3-9. Properties of the PDFIndirectObjectListType class

Name	Type	Multiplicity	Description
Indirect_Object	PDFIndirectObjectType	1..*	The Indirect_Object property captures a single PDF indirect object

			contained in the file.
--	--	--	------------------------

3.10 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 3-10. Properties of the `PDFIndirectObjectType` class

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

3.11 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 3-11. Properties of the `PDFIndirectObjectIDType` class

Name	Type	Multiplicity	Description
Object_Number	<code>cyboxCommon:</code> <code>PositiveIntegerObjectPropertyType</code>	0..1	The <code>Object_Number</code> property captures the number portion of the indirect object ID.
Generation_Number	<code>cyboxCommon:</code> <code>NonNegativeIntegerObjectPropertyType</code>	0..1	The <code>Generation_Number</code> property captures the generation number portion of the indirect object ID.

3.12 `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 3-12. Properties of the `PDFIndirectObjectContentsType` class

Name	Type	Multiplicity	Description
Non_Stream_Contents	<code>cyboxCommon:</code> <code>StringObjectPropertyType</code>	0..1	The <code>Non_Stream_Contents</code> 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.
Stream_Contents	<code>PDFStreamType</code>	0..1	The <code>Stream_Contents</code> property captures the stream contained within in the PDF indirect object, if applicable.

3.13 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 3-13. Properties of the `PDFStreamType` class

Name	Type	Multiplicity	Description
Raw_Stream	<code>cyboxCommon:StringObjectPropertyType</code>	0..1	The <code>Raw_Stream</code> property captures the raw, undecoded stream (i.e., everything between the 'stream' and 'endstream' keywords), as a hex string.
Raw_Stream_Hashes	<code>cyboxCommon:HashListType</code>	0..1	The <code>Raw_Stream_Hashes</code> property captures any hashes of the raw, undecoded stream.
Decoded_Stream	<code>cyboxCommon:HexBinaryObjectPropertyType</code>	0..1	The <code>Decoded_Stream</code> property captures the decoded stream (i.e., after undoing the specified filters in the correct order) as a hex string.
Decoded_Stream_Hashes	<code>cyboxCommon:HashListType</code>	0..1	The <code>Decoded_Stream_Hashes</code> property captures any hashes of the decoded stream.

3.14 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 3-14. Properties of the `PDFDocumentInformationDictionaryType` class

Name	Type	Multiplicity	Description
Title	cyboxCommon: StringObjectPropertyType	0..1	The Title property captures the title for the PDF Document Information Dictionary and reflects what the content producer thinks the dictionary as a whole should be called. The Title property is typically used by humans to reference a particular dictionary; 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 <code>Trapped</code> property captures a name object indicating whether the document has been modified to include trapping information.
----------------	--	------	---

3.15 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 3-15. Properties of the `PDFXrefEntryListType` class

Name	Type	Multiplicity	Description
Cross_Reference_Entry	<code>PDFXrefEntryType</code>	1..*	The <code>Cross_Reference_Entry</code> property captures a single cross-reference subsection entry in the list.

3.16 PDFXrefEntryType Class

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

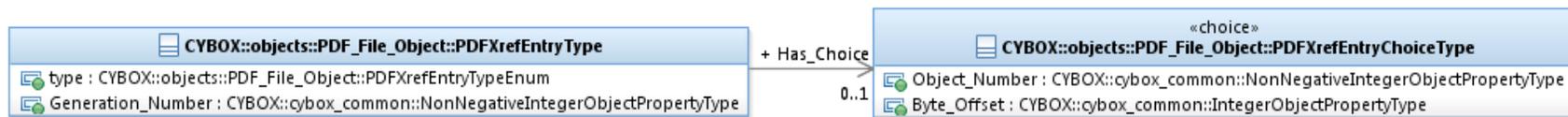


Figure 3-5. UML diagram of the `PDFXrefEntryType` class

The property table of the `PDFXrefEntryType` class is given in [Table 3-16](#).

Table 3-16. Properties of the `PDFXrefEntryType` class

Name	Type	Multiplicity	Description
------	------	--------------	-------------

type	PDFXrefEntryTypeEnum	0..1	The <code>type</code> property specifies the type of the cross-reference entry.
Generation_Number	cyboxCommon: NonNegativeIntegerObjectPropertyType	0..1	The <code>Generation_Number</code> property specifies the 5-digit generation number to be used when an object with the same object number is created.
Has_Choice	PDFXrefEntryChoiceType	1	<p>The <code>Has_Choice</code> property is associated with the class <code>PDFXrefEntryChoiceType</code>. It indicates that there is a choice between the <code>Byte_Offset</code> property or the <code>Object_Number</code> property.</p> <p>Only one of the properties of <code>PDFXrefEntryChoiceType</code> class can be populated at any time. See Section 3.16 for more detail.</p>

The `PDFXrefEntryChoiceType` class is the type of the `Has_Choice` property. In the UML model, this class is associated with the <<choice>> UML stereotype, which specifies that only one of the available properties of the `PDFXrefEntryChoiceType` class can be populated at any time. The property table of the `PDFXrefEntryChoiceType` class is given in [Table 3-17](#).

Table 3-17. Properties of the `PDFXrefEntryChoiceType` class

Name	Type	Multiplicity	Description
Byte_Offset	cyboxCommon: IntegerObjectPropertyType	0..1	<p>The <code>Byte_Offset</code> 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.</p> <p>The <code>Byte_Offset</code> and <code>Object_Number</code> properties MUST NOT both have a value.</p>

Object_Number	cyboxCommon: NonNegativeIntegerObjectPropertyType	0..1	The Object_Number property specifies the 10-digit object number of the next free object. The Byte_Offset and Object_Number properties MUST NOT both have a value.
----------------------	--	------	--

3.17 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 it.

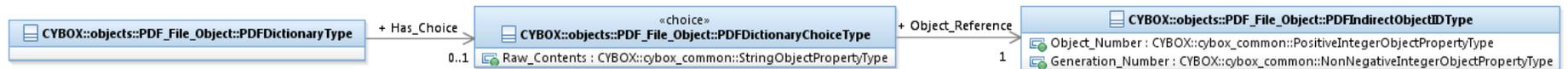


Figure 3-6. UML diagram of the PDFDictionaryType class

The property table of the PDFDictionaryType class is given in Table 3-18.

Table 3-18. Properties of the PDFDictionaryType class

Name	Type	Multiplicity	Description
Has_Choice	PDFDictionaryChoiceType	0..1	The Has_Choice property is associated with the class PDFDictionaryChoiceType. It indicates that there is a choice between the Object_Reference property or the Raw_Contents property. Only one of the properties of PDFDictionaryChoiceType class can be populated at any time. See Section 3.17 for more detail.

The `PDFDictionaryChoiceType` class is the type of the `Has_Choice` property. In the UML model, this class is associated with the `<<choice>>` UML stereotype, which specifies that only one of the available properties of the `PDFDictionaryChoiceType` class can be populated at any time. The property table of the `PDFDictionaryChoiceType` class is given in [Table 3-19](#).

Table 3-19. Properties of the `PDFDictionaryChoiceType` class

Name	Type	Multiplicity	Description
Object_Reference	<code>PDFIndirectObjectIDType</code>	1	The <code>Object_Reference</code> property captures a reference to an indirect PDF object that contains the dictionary via its object and generation numbers. The <code>Object_Reference</code> and <code>Raw_Contents</code> properties MUST NOT both have a value.
Raw_Contents	<code>cyboxCommon:StringObjectPropertyType</code>	1	The <code>Raw_Contents</code> property captures the contents of the dictionary as a string. The <code>Object_Reference</code> and <code>Raw_Contents</code> properties MUST NOT both have a value.

3.18 PDFFileMetadataType Class

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

The property table of the `PDFFileMetadataType` class is given in [Table 3-20](#).

Table 3-20. Properties of the `PDFFileMetadataType` class

Name	Type	Multiplicity	Description
encrypted	<code>basicDataTypes:Boolean</code>	0..1	The <code>encrypted</code> property specifies whether the PDF

			file is encrypted.
optimized	basicDataTypes:Boolean	0..1	The <code>optimized</code> property specifies whether the PDF file has been optimized.
Document_Information_Dictionary	PDFDocumentInformationDictionaryType	0..1	The <code>Document_Information_Dictionary</code> 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 <code>Number_Of_Indirect_Objects</code> property captures the number of indirect PDF objects contained in the file.
Number_Of_Trailers	cyboxCommon: PositiveIntegerObjectPropertyType	0..1	The <code>Number_Of_Trailers</code> property captures the number of trailers contained in the file.
Number_Of_Cross_Reference_Tables	cyboxCommon: PositiveIntegerObjectPropertyType	0..1	The <code>Number_Of_Cross_Reference_Tables</code> property captures the number of cross-reference tables contained in the file.
Keyword_Counts	PDFKeywordCountsType	0..1	The <code>Keyword_Counts</code> property captures the counts of various PDF keyword names in the file.

3.19 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 3-21](#).

Table 3-21. Properties of the `PDFKeywordCountsType` class

Name	Type	Multiplicity	Description
Page_Count	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	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	PDFKeywordCountType	0..1	The ObjStm_Count property captures the number of occurrences of the '/ObjStm' keyword in the PDF file.
JS_Count	PDFKeywordCountType	0..1	The JS_Count property captures the number of occurrences of the '/JS' keyword in the PDF file.
JavaScript_Count	PDFKeywordCountType	0..1	The JavaScript_Count property captures the number of occurrences of the '/JavaScript' keyword in the PDF file.
AA_Count	PDFKeywordCountType	0..1	The AA_Count property captures the number of occurrences of the '/AA' keyword in the PDF file.
OpenAction_Count	PDFKeywordCountType	0..1	The OpenAction_Count property captures the number of occurrences of the '/OpenAction' keyword in the PDF file.
ASCIISHexDecode_Count	PDFKeywordCountType	0..1	The ASCIIHexDecode_Count property captures the number of occurrences of the '/ASCIIHexDecode' keyword in the PDF file.
ASCII85Decode_Count	PDFKeywordCountType	0..1	The ASCII85Decode_Count property captures the number of occurrences of the '/ASCII85Decode' keyword in the PDF file.
LZWDecode_Count	PDFKeywordCountType	0..1	The LZWDecode_Count property captures the number of occurrences

			of the '/LZWDecode' keyword in the PDF file.
FlateDecode_Count	PDFKeywordCountType	0..1	The FlateDecode_Count property captures the number of occurrences of the '/FlateDecode' keyword in the PDF file.
RunLengthDecode_Count	PDFKeywordCountType	0..1	The RunLengthDecode_Count property captures the number of occurrences of the '/RunLengthDecode' keyword in the PDF file.
JBIG2Decode_Count	PDFKeywordCountType	0..1	The JBIG2Decode_Count property captures the number of occurrences of the '/JBIG2Decode' keyword in the PDF file.
DCTDecode_Count	PDFKeywordCountType	0..1	The DCTDecode_Count property captures the number of occurrences of the '/DCTDecode' keyword in the PDF file.
RichMedia_Count	PDFKeywordCountType	0..1	The RichMedia_Count property captures the number of occurrences of the '/RichMedia' keyword in the PDF file.
CCITTFaxDecode_Count	PDFKeywordCountType	0..1	The CCITTFaxDecode_Count property captures the number of occurrences of the '/CCITTFaxDecode' keyword in the PDF file.
Launch_Count	PDFKeywordCountType	0..1	The Launch_Count property captures the number of occurrences of the '/Launch' keyword in the PDF file.
XFA_Count	PDFKeywordCountType	0..1	The XFA_Count property captures the number of occurrences of the '/XFA' keyword in the PDF file.

3.20 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 3-22](#).

Table 3-22. 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.

3.21 PDFObjectTypeEnum Enumeration

The literals of the PDFObjectTypeEnum enumeration are given in [Table 3-23](#).

Table 3-23. Literals of the PDFObjectTypeEnum enumeration

Enumeration Literal	Description
Boolean	
Integer	
String	
Name	
Array	

Dictionary	
Stream	
Null	

3.22 PDFXrefEntryTypeEnum Enumeration

The literals of the `PDFXrefEntryTypeEnum` enumeration are given in [Table 3-24](#).

Table 3-24. Literals of the `PDFXrefEntryTypeEnum` enumeration

Enumeration Literal	Description
In-Use	
Free	

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 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.

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

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

Kenichi Terashita
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

Rutger Prins
Andrei Sirghi
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.

Charles Schmidt
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

James Cabral

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

Clayton Long
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

Chris Roblee

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