



**International  
Standard**

**ISO 14289-2**

**Document management  
applications — Electronic document  
file format enhancement for  
accessibility —**

**Part 2:  
Use of ISO 32000-2 (PDF/UA-2)**

*Applications de la gestion de documents — Amélioration de  
format du fichier du document électronique pour l'accessibilité —*

*Partie 2: Utilisation de l'ISO 32000-2 (PDF/UA-2)*

**First edition  
2024-03**

STANDARDSISO.COM : Click to view the full PDF of ISO 14289-2:2024



**COPYRIGHT PROTECTED DOCUMENT**

© ISO 2024

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
CP 401 • Ch. de Blandonnet 8  
CH-1214 Vernier, Geneva  
Phone: +41 22 749 01 11  
Email: [copyright@iso.org](mailto:copyright@iso.org)  
Website: [www.iso.org](http://www.iso.org)

Published in Switzerland

# Contents

Page

<b>Foreword</b>	<b>v</b>
<b>Introduction</b>	<b>vi</b>
<b>1 Scope</b>	<b>1</b>
<b>2 Normative references</b>	<b>1</b>
<b>3 Terms and definitions</b>	<b>1</b>
<b>4 Notation</b>	<b>2</b>
<b>5 Version identification</b>	<b>2</b>
<b>6 Conformity requirements</b>	<b>3</b>
6.1 General	3
6.2 Conforming files	3
<b>7 Accessible PDF</b>	<b>4</b>
7.1 Principles	4
7.2 Using other standards in the PDF/UA context	5
7.2.1 Accessibility standards	5
7.2.2 Declarations	5
7.2.3 Other PDF subset standards	5
<b>8 File format requirements</b>	<b>5</b>
8.1 General	5
8.2 Logical structure	6
8.2.1 General	6
8.2.2 Real content	6
8.2.3 Logical content order	7
8.2.4 Structure types	7
8.2.5 Additional requirements for specific structure types	8
8.2.6 Structure attributes	18
8.3 Artifacts	19
8.3.1 General	19
8.3.2 Use of the Artifact structure element	19
8.4 Text representation for content	20
8.4.1 General	20
8.4.2 Unicode mapping in tagged PDF	20
8.4.3 Replacements and alternatives for text	20
8.4.4 Declaring natural language	20
8.4.5 Fonts	20
8.5 Real content without textual semantics	23
8.5.1 General	23
8.5.2 Use of ActualText	23
8.6 Text string objects	24
8.7 Optional content	24
8.8 Intra-document destinations	24
8.9 Annotations	24
8.9.1 General	24
8.9.2 Semantics and content	25
8.9.3 Order	29
8.9.4 Alternative descriptions	30
8.10 Forms	30
8.10.1 General	30
8.10.2 Context	31
8.10.3 Fields	32
8.10.4 Non-interactive forms	33
8.11 Metadata	34
8.11.1 General	34

8.11.2	Interactive aspects .....	34
8.12	Navigation .....	34
8.12.1	General .....	34
8.12.2	Outlines .....	34
8.12.3	Page labels .....	34
8.12.4	Article threads .....	34
8.13	Actions .....	34
8.13.1	General .....	34
8.13.2	ECMAScript actions .....	34
8.13.3	URI actions .....	35
8.14	Use of embedded files .....	35
8.14.1	Descriptions for embedded files .....	35
8.14.2	Use of associated files .....	35
<b>Annex A</b>	<b>(informative) Example PDF Declaration .....</b>	<b>36</b>
<b>Annex B</b>	<b>(informative) Examples of attribute usage .....</b>	<b>37</b>
<b>Bibliography</b>	<b>.....</b>	<b>42</b>

STANDARDSISO.COM : Click to view the full PDF of ISO 14289-2:2024

## Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see [www.iso.org/directives](http://www.iso.org/directives)).

ISO draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at [www.iso.org/patents](http://www.iso.org/patents). ISO shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see [www.iso.org/iso/foreword.html](http://www.iso.org/iso/foreword.html).

This document was prepared by Technical Committee ISO/TC 171, *Document management applications*, Subcommittee SC 2, *Document file formats, EDMS systems and authenticity of information*.

A list of all parts in the ISO 14289 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at [www.iso.org/members.html](http://www.iso.org/members.html).

## Introduction

The Portable Document Format (PDF) is a digital format for representing documents. PDF files can be created natively in PDF form, converted from other electronic formats, or digitized from paper. Businesses, governments, libraries, archives, and other institutions and individuals around the world use PDF to represent considerable bodies of important information. These PDF files should be made accessible to users with disabilities.

The primary purpose of the ISO 14289 series, known as PDF/UA, is to define how electronic files in the PDF format can be represented in a manner that allows them to be accessible. The accessibility of a PDF document depends on the inclusion of a variety of semantic information such as (but not limited to):

- machine-readable text presented in a declared language;
- appropriate semantic structures for elements such as paragraphs, lists, tables and headings;
- the organization of semantic structures in logical reading order;
- descriptive metadata, such as alternate descriptions for images.

This document is intended as a companion standard to be used in conjunction with ISO 32000-2 (PDF 2.0). Other standards can also be relevant for the purpose of achieving accessibility.

This document does not replace ISO 14289-1. ISO 14289-1 provides normative guidance based on ISO 32000-1; this document provides normative guidance based on ISO 32000-2.

STANDARDSISO.COM : Click to view the full PDF of ISO 14289-2:2024

# Document management applications — Electronic document file format enhancement for accessibility —

## Part 2: Use of ISO 32000-2 (PDF/UA-2)

### 1 Scope

This document specifies how to use the Portable Document Format (PDF) specified in ISO 32000-2 to construct accessible digital documents using PDF technology.

This document does not specify:

- processes for converting paper or electronic documents to the PDF/UA format;
- technical design, user interface, implementation, or operational details of presentation;
- physical methods of storing these documents such as media and storage conditions;
- required computer hardware and/or operating systems;
- requirements specific to content (beyond facilitating programmatic access and textual representation);
- requirements applying to specific classes (e.g. invoices, reports, etc.) of documents.

### 2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 14289-1, *Document management applications — Electronic document file format enhancement for accessibility — Part 1: Use of ISO 32000-1 (PDF/UA-1)*

ISO 32000-2:2020, *Document management — Portable document format — Part 2: PDF 2.0*

ISO/TS 32005, *Document management — Portable Document Format — PDF 1.7 and 2.0 structure namespace inclusion in ISO 32000-2*

W3C *Digital Publishing WAI-ARIA Module 1.0 (DPUB-ARIA)*, W3C Recommendation, 14 December 2017, <https://www.w3.org/TR/dpub-aria-1.0/>

PDF Association, *PDF Declarations*, September 5, 2019, <https://www.pdfa.org/resource/pdf-declarations/>

### 3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://www.electropedia.org/>

### 3.1 assistive technology AT

hardware or software added to, or incorporated within, a system that increases accessibility for an individual

[SOURCE: ISO 9241-171:2008, 3.5, modified — the example has been removed.]

### 3.2 PDF 1.7 namespace

standard structure namespace for PDF 1.7

[SOURCE: ISO/TS 32005:2023, 3.1]

### 3.3 PDF 2.0 namespace

standard structure namespace for PDF 2.0

[SOURCE: ISO/TS 32005:2023, 3.2]

### 3.4 real content

graphics objects, annotations and form fields representing material intentionally introduced by a document's author and necessary for an *assistive technology* ([3.1](#)) user to understand that document's content

Note 1 to entry: Text, images and paths are all examples of "graphics objects" in PDF files.

Note 2 to entry: The purpose of distinguishing between real and other content is to identify specific objects that the author intended for consumption by a human.

### 3.5 structure attribute

entry containing additional information contained within an attribute object

Note 1 to entry: This term is introduced in ISO 32000-2:2020, 14.7.6.

### 3.6 unique PDF 1.7 element

standard structure element whose type is defined solely in the *PDF 1.7 namespace* ([3.2](#))

[SOURCE: ISO/TS 32005:2023, 3.5]

### 3.7 artifact marked content sequence

artifact defined solely by a marked content sequence, and not defined by a structure element

## 4 Notation

Token characters, text string characters, PDF operators, operands, PDF keywords, the names of keys in PDF dictionaries, and other predefined names are written according to the same convention in ISO 32000-2:2020, Clause 4, and as defined in ISO 32000-2:2020, 7.2 and 7.9.2 (token characters and text string characters, respectively).

## 5 Version identification

The PDF/UA version of a file shall be specified in the value of the **Metadata** entry in the document catalog dictionary using the PDF/UA identification schema defined in this clause.



[Table 1](#) defines the PDF/UA identifier using the namespace URI “<http://www.aiim.org/pdfua/ns/id/>”. The required schema namespace prefix is **pdfuaid**.

NOTE Namespace URIs are intended to be unique strings, and not expected to be resolved as URLs. This URI is retained for compatibility reasons.

**Table 1 — PDF/UA identification schema**

Property	Value type	Category	Description
<b>pdfuaid:part</b>	<i>Open choice of integer</i>	Internal	(Required) PDF/UA part identifier
<b>pdfuaid:rev</b>	<i>Open choice of integer</i>	Internal	(Required) Four-digit year of the date of publication or revision

The value of **pdfuaid:part** shall be 2 for PDF files conforming with this document.

The value of **pdfuaid:rev** shall be the four digits of the year of publication or revision.

The values of the **pdfuaid:part** and **pdfuaid:rev** properties do not determine conformity with this document by themselves; these criteria are specified in [Clause 6](#).

## 6 Conformity requirements

### 6.1 General

The ISO 14289 series define a file format for representing electronic documents commonly known as “PDF/UA”. This document addresses the accessibility of PDF files conforming to ISO 32000-2 (PDF 2.0).

NOTE 1 The proper mechanism by which a file can presumptively identify itself as being a PDF/UA file of a given conformity level is described in [Clause 5](#).

NOTE 2 Regardless of the version number present in the conforming file, conformity requirements specified in this document are based on ISO 32000-2 (PDF 2.0).

### 6.2 Conforming files

Beyond the requirements in [Clause 8](#) and PDF/UA version identification as defined in [Clause 5](#), conformity with this document requires that:

- a file shall adhere to all requirements of ISO 32000-2;
- a file shall adhere to all requirements of ISO/TS 32005;
- a file should not contain any feature that is deprecated in ISO 32000-2;
- a file may include any valid ISO 32000-2 feature that is not explicitly forbidden by this document;
- a file should not contain features described in specifications prior to ISO 32000-2 which are not explicitly defined or identified in ISO 32000-2 or this document;
- an embedded file, if necessary to the understanding of the document, shall be accessible according to objectively verifiable standards, e.g. WCAG 2.2.<sup>[5]</sup> If such an embedded file is a PDF file, it shall conform to the ISO 14289 series.

NOTE 1 Embedded files referenced from the containing file, for use by a human reader, can be necessary to the understanding of the document. See [8.14](#).

NOTE 2 Discouraging deprecated features aligns this document with ISO 19005-4.

NOTE 3 Conformance with ISO/TS 32005 is required because, while ISO 32000-2 did not deprecate any structure types defined in PDF 1.7, ISO 32000-2:2020, Annex L provided containment rules only for structure types defined in the PDF 2.0 namespace. ISO/TS 32005 expands these containment rules to cover both the PDF 2.0 namespace and those elements unique to the PDF 1.7 namespace. By adhering to the requirements of ISO/TS 32005, it is possible to include both PDF 2.0 and PDF 1.7 structure types in files that conform to this document.

A file may declare its support for and conformity with PDF extensions to ISO 32000-2:2020 by use of an Extensions dictionary as specified in ISO 32000-2:2020, 7.12.

NOTE 4 Use of extensions can potentially negatively impact accessibility, compatibility and consistency of files when consumed by different processors.

Although other accessibility standards provide guidance on accessible content, this document deliberately avoids requirements related to content. Conformity to PDF/UA alone does not ensure that the content of a document is accessible. Cases not covered by PDF/UA include:

- where authors have used colour or contrast in an inaccessible manner;
- where ECMAScript present in the file can generate inaccessible results;
- where text content is potentially inaccessible to those with certain cognitive impairments;
- whether a particular content item is real content or artifact content.

NOTE 5 Requirements governing author's creative choices, (e.g. requirements for plain language), or the use of specific degrees of colour-contrast, are out of scope for this document. For such requirements, authors can refer to WCAG 2.2<sup>[5]</sup> and other applicable standards such as ISO 24495-1.

## 7 Accessible PDF

### 7.1 Principles

The degree to which document content can be described as accessible depends on the extent to which the semantic significance and context of each element of content is readily available to as many people and PDF processors as possible.

Accessible PDF files adhere to the RISE principles:

- **Reliable:** Files comply to the greatest possible extent with applicable specifications to facilitate robustness in processing across a wide range of use cases and software.
- **Interoperable:** Elements of content can be readily exchanged between computing systems without loss of semantics.
- **Suitable:** Suitability for the widest range of PDF processing applications and use-cases within the intended user population, taking account of special abilities, variations in capabilities, diversity in tasks, and differing environmental, economic and social circumstances.
- **Equitable:** Equitable solutions provide the same means of use for all users: identical whenever possible; equivalent when not.

NOTE These principles have been adapted from ISO 9241-171:2008 and are consistent with WCAG's POUR principles.<sup>[5]</sup>

In PDF, access to content is provided by the structures of the file, the software reading the file, and by the assistive technology consuming the software's output. The content of a document is considered accessible if real content as specified in ISO 32000-2:2020, 14.8.2.2 is represented with text or has a text-based equivalent presented in logical reading order with appropriate semantics. This document establishes how to apply these principles to a PDF file; it addresses the way content is encoded within a conforming PDF/UA file but does not regulate the document's content itself, as seen in the following examples.

EXAMPLE 1 The maximum number of characters in a line of text is unregulated.

EXAMPLE 2 Specific colours, colour-combinations, or contrast ratios are unregulated.

EXAMPLE 3 The choice to use plain language or other language is unregulated.

EXAMPLE 4 The size of fonts is unregulated.

## 7.2 Using other standards in the PDF/UA context

### 7.2.1 Accessibility standards

Diverse content requirements (e.g. the examples provided in [7.1](#)) can be addressed with other specifications. [7.2.2](#) describes a means of claiming conformity with one or more such specifications.

### 7.2.2 Declarations

Many standards and regulations related to accessibility go beyond the requirements of PDF/UA. To facilitate claims of conformity with such standards, a PDF/UA conforming document may, in addition to the PDF/UA identification schema specified in [Clause 5](#), include PDF Declarations. An example PDF Declaration is provided in [Annex A](#).

EXAMPLE 1 A PDF Declaration can serve as the author's statement attesting that:

- all of a given file's content conforms to a specific accessibility standard (such as WCAG 2.2, Level AA);<sup>[5]</sup>
- all images and figures have been checked for appropriate alternate descriptions;
- the file has been certified to be accessible when using a specific PDF/UA processor.

Files in conformity with this document may leverage the PDF Declarations mechanism to include conformity claims for the entire file, for portions of the file (even specific objects), or both.

EXAMPLE 2 A PDF Declaration can serve as the author's statement attesting that an individual movie embedded in a PDF/UA file conforms to WCAG 2.2 Level AA.<sup>[5]</sup>

NOTE 1 It is expected that the file, portion, or object to which a PDF Declaration applies fully conforms to the standards to which they are declaring conformity.

NOTE 2 The PDF Association's list of PDF Declarations is fully extensible. Other PDF Declarations can be defined there and elsewhere.

An embedded file that is not PDF, if necessary to the understanding of the document, shall have an associated PDF Declaration in accordance with PDF Declarations, indicating its conformance to an appropriate accessibility standard. The PDF Declaration, as specified in PDF Declarations, shall be included through the Metadata entry in its file specification dictionary. Any embedded files may have PDF Declarations associated with them.

### 7.2.3 Other PDF subset standards

A file in conformity with this document may also conform to other ISO subset standards based on ISO 32000-2 (PDF 2.0) including ISO 19005-4 (PDF/A-4) and ISO 15930-9 (PDF/X-6).

## 8 File format requirements

### 8.1 General

Conformity with this document imposes requirements and restrictions beyond those specified in ISO 32000-2 and ISO/TS 32005. [Clause 8](#) specifies these additional requirements and restrictions.

Beyond requirements for aspects of logical structure, consistent and accurate rendering is also important to meet the objectives of this document. Accordingly, [8.4.5](#) includes requirements matching those of the corresponding clause in ISO 19005-4 (PDF/A-4) and ISO 15930-9 (PDF/X-6).

## 8.2 Logical structure

### 8.2.1 General

Conformity with this document depends on the correct use of semantically appropriate structure elements enclosing the document's content, along with the structure elements' properties and structure attributes (additional information attached to the structure element).

### 8.2.2 Real content

All real content, as specified in ISO 32000-2:2020, 14.8.2.2, shall be enclosed within semantically appropriate structure elements using appropriate attributes according to the definitions for each structure type (name objects that identify the nature of the structure element), as defined in ISO 32000-2:2020, 14.8 and ISO/TS 32005.

NOTE 1 The concept of real content is defined in ISO 32000-2:2020, 14.8.2.2.1.

NOTE 2 ISO/TS 32005 defines the permitted PDF 1.7 structure elements and their inclusion rules in relation to the PDF 2.0 structure elements as defined in ISO 32000-2.

NOTE 3 Some structure elements with types such as **Div** and **Span** have no intrinsic semantics. **Div** is used to apply attributes to groups of structure elements and their descendants (including their respective contents), while **Span** is used to apply attributes to content.

NOTE 4 Whether attributes are appropriate depends on context and content properties (see [8.2.6](#)).

EXAMPLE 1 The **Placement** attribute is appropriate on a **Figure** structure element whose content is inline in a paragraph.

EXAMPLE 2 The **Placement** attribute is optional when a **Figure** is placed in a block flow between two paragraphs.

EXAMPLE 3 The **Placement** attribute with a value of *Block* is appropriate for an **FENote** structure element placed in the block flow, whereas the **Placement** attribute with a value of *Inline* is appropriate for an **FENote** occurring inline contained within a **P** structure element.

EXAMPLE 4 The **TextDecorationType** attribute with a value of *LineThrough* is appropriate for a structure element enclosing struck text.

EXAMPLE 5 The **TextPosition** attribute with a value of *Sup* is appropriate for superscript characters.

Tagging shall reflect the semantics of the document's real content regardless of how the real content was laid out, paginated, or encoded by an authoring tool. Real content that is typically represented by a single semantic structure element with intrinsic semantics shall not be represented by several such structure elements of the same structure type.

EXAMPLE 6 A single table spanning two or more pages is correctly enclosed in a single **Table** structure element.

NOTE 5 It is incorrect to have a single table spanning two or more pages enclosed in multiple **Table** structure elements.

EXAMPLE 7 A single paragraph that spans two or more pages or columns is correctly enclosed within a single **P** structure element.

NOTE 6 It is incorrect to have multiple lines of a single paragraph enclosed within two or more **P** structure elements.

NOTE 7 How content is encoded within a content stream does not indicate the semantically appropriate structure element(s). Unlike ISO 14289-1, this document clearly specifies that the use of images or vector-based drawings does not always require a **Figure** structure element.

EXAMPLE 8 An image solely used to represent text with no illustrative purpose can be enclosed in a **Span** structure element with appropriate **ActualText**, as opposed to a **Figure** structure element.

Content that is not considered real content shall be an artifact in accordance with [8.3](#).

### 8.2.3 Logical content order

The logical content order of structure elements and their contents according to ISO 32000-2:2020, 14.8.2.5 shall be semantically correct.

NOTE 1 This requirement is equivalent to WCAG 2.2, Success Criterion 1.3.2.[\[5\]](#)

NOTE 2 This document does not impose any understanding of what “semantically correct” logical content order means to any given author. Two common example approaches to ordering real content are to follow the order of the geographical page layout appropriate to the language or script (e.g. top-down, left-right in some languages), or to order the most important real content first. The author chooses the approach and thus establishes the intended logical content order for the document.

EXAMPLE 1 The correct logical content order, in a trifold brochure in English, with its title on the right-hand panel of a landscape page, would typically locate the title content before the rest of the brochure in the structure tree.

Artifact content intended to be consumed in a single unit shall be enclosed within a single **Artifact** structure element or artifact marked content sequence.

NOTE 3 **Artifact** structure elements and artifact marked content sequences are specified in ISO 32000-2:2020, 14.8.2.2.2.

EXAMPLE 2 Textual content in a page footer would usually be identified as a single artifact.

EXAMPLE 3 Page numbers.

If artifact content has a semantic order, the ordering of that content within the artifact shall match the semantic order.

In some artifacts, some of the content has semantic order and some does not. Some artifact content, particularly drawing content, does not have a semantic order.

EXAMPLE 4 Paths comprising table borders have no semantic order.

EXAMPLE 5 Page numbers with a background have a partial semantic order in that the order of the digits of the page number is semantic, but the relative order of the background to the page number is not.

### 8.2.4 Structure types

All structure elements shall belong to, or be role mapped to, at least one of the following namespaces specified in ISO 32000-2:2020, 14.8.6:

- the PDF 1.7 namespace;
- the PDF 2.0 namespace;
- the MathML namespace.

Other namespaces may be used by files in conformity with this document, but all structure elements in such namespaces shall have their structure types role mapped as described in this subclause. Such role mapping may be transitive through other namespaces.

A structure element with no explicit namespace may be present. Such a structure element shall have, after any role mapping, a structure type matching one of the unique PDF 1.7 element types and as restricted by this document.

NOTE 1 The default standard structure namespace in ISO 32000-2 is defined as the PDF 1.7 namespace.

NOTE 2 [8.2.5.14](#) restricts the use of the **Note** structure type from the default standard structure namespace, and it is therefore not permitted despite being a unique PDF 1.7 element.

Within a given explicitly provided namespace, structure types shall not be role mapped to other structure types in the same namespace.

If custom structure types are role mapped, either directly or transitively, to standard structure types, they shall conform to the requirements for the standard structure types to which they are role mapped, including the containment rules specified in ISO/TS 32005:2023, Clause 7.

**EXAMPLE** A custom structure type mapped to a **Caption** standard structure type would incur all requirements for the **Caption** structure type, including its permitted location in the structure tree with respect to the structure element enclosing the captioned content and the element's permitted descendants.

## 8.2.5 Additional requirements for specific structure types

### 8.2.5.1 General

ISO 32000-2:2020, 14.8.4 and ISO/TS 32005:2023, 5.5 describe the full list of standard structure types for the PDF 2.0 namespace and PDF 1.7 namespace, respectively, and provide both requirements and examples of their use.

Usage of the standard structure types shall be in accordance with the requirements specified in both ISO 32000-2:2020, 14.7.4.2 and ISO/TS 32005 for the PDF 2.0 namespace and PDF 1.7 namespace.

[8.2.5.2](#) to [8.2.5.33](#) provide additional requirements for specific structure types.

**NOTE** [8.2.5.2](#) to [8.2.5.33](#) appear in the order in which the structure types occur in ISO 32000-2. If a structure type does not appear in [8.2.5.2](#) to [8.2.5.33](#), then no additional requirements apply beyond those specified in ISO 32000-2 and ISO/TS 32005.

Requirements specific to structure attributes are addressed in [8.2.6](#).

### 8.2.5.2 Document and DocumentFragment

The structure tree root (defined in ISO 32000-2:2020, 14.7.2) shall contain a single **Document** structure element as its only child, as specified in ISO 32000-2:2020, Annex L and ISO/TS 32005. The namespace for that element shall be specified as the PDF 2.0 namespace, in accordance with ISO 32000-2:2020, 14.8.6.

**NOTE 1** PDF/UA-2-conforming files cannot solely conform to PDF 1.7 because this subclause requires that the **Document** structure element is in the PDF 2.0 namespace.

When a document encloses content from another document that is either a logical document or logical document fragment as specified in ISO 32000-2:2020, 14.8.4.3, that content should be enclosed in either a **Document** structure element or a **DocumentFragment** structure element, respectively.

The **DocumentFragment** structure type shall only be used when the author's intent is to identify real content as originating in another document.

**EXAMPLE** Excerpts from documents included inside another document.

**NOTE 2** **DocumentFragment** is inappropriate for content (e.g. some quotations) whose semantic intent does not rely on identification as an extract from another document.

**NOTE 3** Because the content is a fragment of another document, it is possible that heading levels do not align with the overall document and would not be expected to impact subsequent heading levels. However, headings in that fragment would be self-consistent and not multiple subsets of another document, as described in ISO 32000-2:2020, 14.8.4.3.

### 8.2.5.3 Part

When a grouping of content has semantic purpose unrelated to the document's headings hierarchy, it should be enclosed within a **Part** structure element. ISO 32000-2:2020, Table 365 provides examples distinguishing the use of **Part** from the use of the **Sect** structure type.



#### 8.2.5.4 Article (Art)

Where content is semantically representative of a self-contained article in a larger document that contains multiple such articles, that content shall be enclosed in an **Art** structure element.

NOTE As such, articles often include titles, their own heading levels, and sub-structure.

Where a title applies to an entire article, the **Title** structure element shall be included inside the **Art** structure element, either directly or indirectly.

#### 8.2.5.5 Section (Sect)

When a document is divided into sections, **Sect** structure element(s) should be used to contain the structure elements and real content that comprise such sections. When a heading applies to an entire section, then that heading structure element shall be contained within that **Sect** structure element.

NOTE 1 A section is a thematic grouping of content and structure elements, typically with a heading structure element at its start.

NOTE 2 The presence of the **Sect** structure element enables efficiently skipping sections of a document.

#### 8.2.5.6 Division (Div)

Where attributes or properties are common to a set of elements, a **Div** structure element may be used to apply those common attributes or properties to that group of elements.

EXAMPLE A set of paragraphs in a common language that differs from that of the surrounding content.

NOTE A structure element with the **Div** structure type provides no direct semantics of its own.

#### 8.2.5.7 Block quotation (BlockQuote)

**BlockQuote** shall be used to identify groups of block-level content that is quoted from another source other than the surrounding content. For inline-level content, refer to **Quote** (8.2.5.18).

NOTE 1 The other source can be a different author but can also be a different article in the same document or part of a different document by the same author.

NOTE 2 Pull-quotes are duplicative and are therefore not considered to be block quotes but are generally considered asides.

#### 8.2.5.8 Table of Contents (TOC/TOCI)

Tables of contents shall be identified using the **TOC** and **TOCI** standard structure types as defined in ISO/TS 32005:2023, Table 2.

NOTE This requirement clarifies the requirement in 8.2.2 because the presence of **TOC** and **TOCI** in the PDF 1.7 namespace, but not in the PDF 2.0 namespace, can cause confusion regarding appropriate use of the structure types specific to tables of contents.

Each **TOCI** in the table of contents shall identify the target of the reference using the **Ref** entry, either directly on the **TOCI** structure element itself or on a child structure element contained within, such as a **Reference** structure element.

Leaders shall be identified as artifacts using one of the methods described in 8.3.

EXAMPLE Leaders are commonly represented by a series of dots (e.g. ".....") separating the item and corresponding page number.

If using an **Artifact** structure element to enclose leaders, the location of such entries in the structure tree shall follow appropriate logical ordering and containment.

### 8.2.5.9 Aside

The **Aside** structure type shall be used to enclose real content that occurs outside the main flow of the content.

EXAMPLE 1 A sidebar containing advertising on a magazine page is typically understood as outside the main flow of the content.

EXAMPLE 2 Side notes in a textbook.

If content in an **Aside** is related to the main flow of content, the parent of the **Aside** structure element shall be the deepest structure element to which the **Aside** structure element's content relates.

NOTE **Aside** loosely couples its content to a single location in the structure tree. Refer to [8.2.5.14](#) for information about the related **FENote** structure element.

### 8.2.5.10 NonStruct

The **NonStruct** structure type may be used as the role mapping target of a structure element with a custom structure type in a custom namespace to indicate that the structure elements of this type and their attributes are not relevant. Descendant structure elements and their respective content retain their semantics.

EXAMPLE A custom namespace can have a structure type representing pauses in spoken language. In such cases, **NonStruct** can be used as a role mapping target to indicate the lack of semantic significance of that custom type in relation to the standard structure namespaces.

### 8.2.5.11 Paragraph (P)

The **P** structure type shall be used for any content that is considered a paragraph, regardless of the containing parent structure element and the number of paragraphs within a parent structure element. The **P** structure type should only be used when content is intended to be a semantic paragraph, or when there is not a more semantically appropriate structure type. Content that is not intended to be a semantic paragraph (e.g. a fragment of text in the **LBody** of a list item) should not be enclosed in a **P**.

EXAMPLE 1 In a list item, where the content enclosed within the **LBody** is a paragraph, then that content is tagged as a paragraph.

When more than one paragraph exists within a parent element, each paragraph shall be tagged with a **P** structure element.

EXAMPLE 2 Where a footnote contains multiple paragraphs as its content, each paragraph is enclosed within its own **P** structure element.

Where textual real content is insufficiently semantically defined by its parent element (or enclosing ancestor elements), a **P** structure element may be used to separate that content from surrounding content and to provide a base semantics for the text.

NOTE See examples in [8.2.2](#).

### 8.2.5.12 Heading (Hn) and (H)

Conforming files shall use the explicitly numbered heading structure types (**H1-Hn**) and shall not use the **H** structure type.

NOTE 1 The **H** structure type requires processors to track section depth, which adds an unnecessary burden on processors and can cause ambiguity.

Where a heading's level is evident, the heading level of the structure element enclosing it shall match that heading level.

NOTE 2 Standards such as ISO 14289-1 include requirements on the use of sequential heading levels. This document does not, in accordance with WCAG,<sup>[5]</sup> include such a requirement, but instead focuses on ensuring that correct semantics are provided for a given document's content.



EXAMPLE A heading with the real content “4.5.6 Foobar” is evidently at heading level 3.

### 8.2.5.13 Title

Titles shall be identified by the **Title** standard structure type and shall not be identified as a heading.

NOTE This requirement is specific to real content and does not pertain to requirements for metadata. As titles can vary depending on exact usage, there is no intent to require a match between real content tagged with **Title** and **dc:title** information in XMP metadata.

### 8.2.5.14 Footnotes and Endnotes (FENote)

#### 8.2.5.14.1 Structure requirements

**FENote** effectively replaces the **Note** structure type specified in ISO 32000-1:2008, 14.8.4.4.1. The **Note** standard structure type shall not be present in conforming files.

NOTE 1 This does not prohibit using a **Note** structure type in any other namespace if it is role mapped to an allowed structure type in the PDF 1.7 namespace or PDF 2.0 namespace.

Real content that refers to footnotes or endnotes (real content enclosed in **FENote** structure elements) shall use the **Ref** entry as specified in 8.8 on the referring structure element to reference the **FENote**. The corresponding **FENote** shall also use the **Ref** entry to identify all citations that reference it. If the reference to the footnote or endnote is interactive, it shall point to the footnote or endnote by means of a link annotation using a structure destination, as specified in ISO 32000-2:2020, 12.3.2.3.

NOTE 2 **FENote** tightly couples its content to one or more locations in the structure tree.

#### 8.2.5.14.2 FENote Attribute Owner

To facilitate disambiguation for the intent of specific **FENote** structure elements, a new attribute owner should be used as defined in Table 2.

Table 2 — FENote Attribute Owner

Owner value for the attribute object's <b>O</b> entry	Description
<b>FENote</b>	Attribute governing type of footnotes, endnotes and other types of <b>FENote</b> .

If appropriate to the content, the attributes described in Table 3 should appear on an **FENote** structure element. These attributes may be defined in attribute objects whose **O** entry has the value *FENote*.

NOTE It is possible that a processor does not know the type of **FENote**, therefore an explicit **NoteType** cannot be required.

Table 3 — Standard FENote attributes

Key	Type	Value
<b>NoteType</b>	name	<p>(Optional) The type of the <b>FENote</b> indicating a more specific semantic definition for the <b>FENote</b>. The value of the <b>NoteType</b> shall be one of the following and shall be applied as described here.</p> <p><i>Footnote</i> A subtype indicating a footnote</p> <p><i>Endnote</i> A subtype indicating an endnote</p> <p><i>None</i> A subtype indicating that the type is not specified</p> <p>Default value: <i>None</i></p>

### 8.2.5.15 Subdivision (Sub)

**Sub** should be used to identify semantic subdivisions within a block-level element.

### 8.2.5.16 Label (Lbl)

Real content that labels other real content shall be enclosed within **Lbl** structure elements.

EXAMPLE 1 Footnotes are often labelled by a superscript number, letter, or symbol.

NOTE 1 Artifact content can label real content. In such a case, a **Lbl** structure element is not appropriate.

EXAMPLE 2 Line numbers in legal documents or source code are usually artifacts, but, when quoted in other texts, line numbers can be real content (i.e. using **Sub** and **Lbl** structure elements to enclose the line and its line number).

To strongly associate a label with the real content being labelled, the **Lbl** shall be a descendant of a structure element that semantically groups both the **Lbl** and the content being labelled. Labels and labelled content may be enclosed within an additional substructure.

NOTE 2 The label is associated with the most immediate semantically significant ancestor element that groups both the label and the content being labelled.

### 8.2.5.17 Span

The **Span** structure type shall be used inline to apply attributes and properties to real content where all of the following are true:

- no other semantically appropriate inline structure element fully encompasses only the relevant content;
- the properties and structure attributes do not apply to the parent structure element;
- the semantics are not otherwise conveyed via properties on marked content sequences.

EXAMPLE In the middle of a paragraph, a word occurs in a different colour and the **Span** element is used to provide the appropriate layout attribute (colour).

A **Span** structure element may be used anywhere it is permitted (see ISO/TS 32005).

### 8.2.5.18 Quote

The **Quote** structure type shall be used to identify inline-level content that is quoted from another source other than the surrounding content. For block-level quotations, see **BlockQuote** (8.2.5.7).

NOTE Content that has quotation marks around it is not necessarily considered a quote. For example, newspapers often use quotation marks to indicate non-factual or interpretive statements, which are not quotes.

### 8.2.5.19 Emphasis (Em) and Strong

The **Em** and **Strong** structure types are used to indicate emphasis and should not be used for other purposes.

NOTE Styling is sometimes used to denote constructs in a document, for example, keywords. The **Em** and **Strong** structure types are not intended for such purposes.

### 8.2.5.20 Link and Reference

A link annotation (see 8.9.2.4.2) and its associated content shall be enclosed in either a **Link** or **Reference** structure element. The **Link** structure type should be used for external links and the **Reference** structure type should be used for intra-document targets. Link annotations that target different locations shall be in separate **Link** or **Reference** structure elements.

NOTE External vs intra-document linkage is a semantic concept and is not intended to describe the syntactic mechanism used to provide such a target. For example, a target that uses a URL to reference the file in which the link originates can be considered an internal target and included in a **Reference** structure element. See 8.8 for additional requirements regarding intra-document destinations.

EXAMPLE 1 A bibliography entry including an ISBN or URN referencing external material can be linked using a **Link** structure element while a link to another page in the same document can be linked using a **Reference** structure element.

Multiple link annotations shall be included in a single **Link** or **Reference** structure element if they target the same location and are semantically perceived to be a single link.

EXAMPLE 2 A URL that is split across two pages has at least two link annotations but is semantically perceived as the same link and tagged within a single **Link** structure element.

EXAMPLE 3 An authoring application can choose to produce two link annotations if content being linked is split across multiple lines and is tagged within a single **Link** or **Reference** structure element.

The **Reference** structure type may be used for both interactive links using link annotations and may also be used for non-interactive content where a reference is present.

EXAMPLE 4 An example of a non-interactive reference is a callout to a table in a document from within a paragraph.

When a **Link** or **Reference** structure element encloses content that requires a description, an **Alt** entry may be used to describe both the content and the link.

EXAMPLE 5 A **Link** structure element with an **Alt** entry enclosing a link annotation and a logo.

#### 8.2.5.21 Annot

Use of the **Annot** structure type shall be in accordance with the requirements of [8.9](#).

#### 8.2.5.22 Form

Use of the **Form** structure type shall be in accordance with the requirements of [8.10](#).

#### 8.2.5.23 Ruby (Ruby, RB, RT, RP)

Ruby is often used to provide glosses or phonetic aids adjacent to the base text to which it applies. When ruby is used for this purpose, the **Ruby** structure element shall be used. The **Ruby** structure element shall contain a single **RB** structure element enclosing the annotated base text, and a single **RT** structure element enclosing the annotating ruby text.

In cases where ruby typesetting is used for other purposes, the **Ruby** structure element shall not be used.

NOTE 1 Interlinear notes typeset in a smaller ruby-like font can be tagged using the **Aside** structure element.

The grouping of **Ruby** structure elements should be such that individual words or other semantically atomic units of content are not subdivided, even when a more granular pairing of ruby text to base text is available.

NOTE 2 Ensuring that the grouping of **Ruby** structure elements respects semantic boundaries can be helpful to assistive technology.

When a portion of the ruby annotation in a ruby pair would otherwise duplicate the base text, the overlapping characters are typically omitted from the ruby text. When tagging such content in a single **Ruby** structure element, **ActualText** shall be used on the **RT** sub-element to provide a complete textual replacement restoring all omitted characters.

EXAMPLE 1 A document contains the Japanese word “文書”, annotated with “ぶんしよ” as a phonetic gloss in ruby. The ruby assembly is tagged as:

```
<Ruby><RB>文書</RB><RT>ぶんしよ</RT></Ruby>
```

even though the readings of the individual characters “文” and “書” are “ぶん” and “しよ”, respectively.

EXAMPLE 2 A document contains the Japanese word “今日”, annotated with “きょう” as a ruby phonetic gloss. This ruby pair cannot be subdivided any further, and as such is tagged as:

```
<Ruby><RB>今日</RB><RT>きょう</RT></Ruby>
```

EXAMPLE 3 When annotating the word “振り仮名” with its reading “ふりがな”, one would ordinarily put “ふ” adjacent to “振”, and “がな” adjacent to “仮名”. The “り” is typically not duplicated in the ruby text. When tagging this pair in a single **Ruby** structure element (e.g. to meet an accessibility need), it is rendered as:

```
<Ruby><RB>振り仮名</RB><RT ActualText="ふりがな">ふ がな</RT></Ruby>
```

EXAMPLE 4 A Japanese-language document contains the English name “International Organization for Standardization” annotated with “国際標準化機構” as a native ruby gloss. Since the base text forms a single semantic unit, this ruby pair is tagged as:

```
<Ruby><RB>International Organization for Standardization</RB><RT>国際標準化機構</RT></Ruby>
```

In some cases, phonetic glosses are rendered as parentheticals in running text, as an alternative to ruby text, but with the same semantics. In these cases, the **Ruby** structure element should be used. When used in this manner, the **Ruby** structure element shall consist of a four-element subsequence: **RB**, **RP**, **RT**, **RP**, with the content distributed as follows:

- the **RB** element shall enclose the annotated portion of the base text;
- the first **RP** element shall enclose the opening delimiter of the annotation text;
- the **RT** element shall enclose the annotation text;
- the second **RP** element shall enclose the closing delimiter of the annotation text.

EXAMPLE 5 Word-processing software without ruby support can annotate the word “振り仮名” as “振り仮名(ふりがな)”. This is tagged as:

```
<Ruby><RB>振り仮名</RB><RP> (</RP><RT>ふりがな</RT><RP>) </RP></Ruby>
```

#### 8.2.5.24 Warichu (WT, WP)

Content typeset as warichu shall be tagged in a three-element sequence consisting of the structure elements **WP**, **WT** and **WP**, grouped inside a **Warichu** structure element. The first **WP** structure element shall enclose the opening delimiter separating the warichu content from the surrounding content. The main body of the warichu content shall be enclosed in a single **WT** structure element. The second **WP** structure element shall enclose the closing delimiter separating the warichu content from the surrounding content.

The **Warichu** structure element shall not be used for content that is not typeset as warichu.

NOTE Warichu notes are semantically similar to footnotes. Some authors use both note variants for different purposes.

#### 8.2.5.25 List (L, LI, LBody)

If a list includes real content used as a label, such labels shall be enclosed within **Lbl** structure elements.

If **Lbl** structure elements are present, the **ListNumbering** attribute (see [Table B.5](#)) shall be present on the respective **L** structure element and should have the value that most closely approximates the intended labelling scheme; in such cases the value *None* shall not be used.

Any real content within an **LI** structure element that is not enclosed in a **Lbl** structure element shall be enclosed in an **LBody** structure element.

If a list continues from a previous list, the **ContinuedList** attribute shall be present and have a value of *true*. If the previous list is present in the document the **ContinuedFrom** attribute shall be present.

NOTE There is value to the **ContinuedList** attribute even when the previous list is not present, since it would help to explain the partial nature of the content, for example, partial numbering.

If a list is continued in a subsequent list, or continues from a previous list, a **Ref** entry should be present and, if present, it shall reference all list segments.

### 8.2.5.26 Table (Table, TR, TH, TD, THead, TBody, TFoot)

Tables shall be regular.

NOTE 1 Tables are regular when the number of logical cells is equal in each row after accounting for **RowSpan** and **ColSpan** attributes.

Row groupings formed by **THead**, **TBody** and **TFoot** structure elements shall be regular.

NOTE 2 Row groups are regular when the number of logical cells is equal in each row after accounting for **RowSpan** and **ColSpan** attributes and where cells do not span row groups.

When a table contains header cells, that table shall provide sufficient semantic information to accurately determine which table header cells (structure element **TH**) pertain to other cells as specified in ISO 32000-2:2020, 14.8.5.7.

NOTE 3 Some tables do not have headers.

ISO 32000-2:2020, 14.8.4.8.3 describes an algorithm to determine header cells for a cell in the table. If the algorithm results in default values for **Scope**, as specified in ISO 32000-2:2020, 14.8.4.8.3, that are insufficient to describe the relationship between the header cell and other cells, then the **Scope** attribute shall be specified.

If the algorithm results in a header cell with an implicit or explicit **Scope** that cannot describe the header/cell relationships for all the cells in the table, then the **Headers** attribute shall be present for all cells to which a header applies (see [B.4](#)).

NOTE 4 This means that if the **Headers** attribute is used anywhere in the table, it is required for all cells in the table that have headers, regardless of whether **Scope** describes their relationships.

NOTE 5 The use of **Headers** does not negate the need for **Scope**.

As noted in ISO 32000-2:2020, Annex L, Table rows (**TR**) may include both **TH** and **TD** cells.

NOTE 6 Tables can contain cells that are column headers, row headers and both column and row headers.

Examples of table-specific attributes are provided in [B.4](#).

NOTE 7 End user reliance on assistive technology implies a need for as much information as possible about the structure of tables. Table header cells and their attributes play a key role in providing structural information.

### 8.2.5.27 Caption

All content that captions other content shall be enclosed within a **Caption** structure element.

When present, the **Caption** element shall be a child of the structure element enclosing the content to which the caption pertains. When the caption is intended to be consumed before the content to which it pertains, it shall be the first child, but when it is intended to follow the content, it shall be the last child.

### 8.2.5.28 Figure

#### 8.2.5.28.1 Figure semantics

As **Figure** structure elements represent fixed-layout content intended to be consumed as a unit, a **Figure** structure element shall enclose all content used to generate the final appearance, including background content.

A **Figure** may contain substructures to semantically describe the graphics objects that comprise the real content's appearance.

NOTE Graphics objects include text, vector paths, raster images and other objects as defined in ISO 32000-2:2020, 8.2.

EXAMPLE 1 An organization chart enclosed by a single **Figure** structure element that itself contains substructure to describe each subunit of the chart.

EXAMPLE 2 A background image intersecting an image which the author considers to be an intrinsic part of the figure is artifact content in this case.

The presence of **Figure** structure elements implies additional requirements in accordance with [8.5](#).

#### 8.2.5.28.2 Figure properties

A **Figure** structure element shall have at least one of the following properties:

- a) an alternate description (**Alt** property), as specified in ISO 32000-2:2020, 14.9.3;
- b) a replacement text (**ActualText** property) that represents the content enclosed by the **Figure** structure element.

**Figure** elements including an **ActualText** property, irrespective of the presence of an **Alt** property, shall be contained within the semantically appropriate block-level element.

NOTE 1 Requirements for **Figure** structure elements including **ActualText** are provided in [8.5.2](#).

NOTE 2 WCAG 2.2, 1.1[5] also has requirements for alternative descriptions.

#### 8.2.5.29 Formula

##### 8.2.5.29.1 Mathematical expressions

Mathematical expressions shall be represented using either structure types defined in presentation MathML, as specified in ISO 32000-2:2020, 14.8.6.3, or as presentation MathML via the Associated Files mechanism specified in ISO 32000-2:2020, 14.13.6, or both.

NOTE 1 It is often easier for authors to represent mathematical expressions with MathML via an Associated File, which is why this document supports using both structure elements and Associated Files to represent math.

The **math** structure type, as defined in ISO 32000-2:2020, 14.8.6.3, shall occur only as a child of a **Formula** structure element. When an associated file is used to provide a representation of a **Formula** in presentation MathML, it shall be associated with the respective **Formula** structure element and shall include an **AFRelationship** entry with the value of *Supplement*.

NOTE 2 In general, MathML can represent the same formula in different ways.

EXAMPLE Using the MathML namespace to tag the mathematics in  $\sqrt[3]{x}$ , the 3 is tagged with MathML's **mn** element, the  $x$  is tagged with **mi**, and the entire root is tagged with **mroot**. Fully tagging every element greatly simplifies the processor's burden to synchronize the highlighting of speech and text.

NOTE 3 Using MathML to represent mathematical expressions allows a processor to:

- speak the expression customized to a user's needs;
- navigate the expression so that larger expressions can be more easily understood;
- convert the expression to a mathematical braille code;
- synchronize speech and highlighting of text;
- perform computations based on mathematical formulae;
- export the equation to MathML and other formats.



### 8.2.5.29.2 Non-mathematical scientific formulae

Non-mathematical scientific formulae shall be enclosed within **Formula** structure elements and shall have either an **Alt** entry or an **ActualText** entry in its structure element dictionary. Such formulae should be represented using either structure types defined in an appropriate namespace or via the Associated Files (ISO 32000-2:2020, 14.13.5) mechanism, or both.

EXAMPLE A chemical formula, such as  $2\text{Na} + 2\text{H}_2\text{O} \rightarrow \text{H}_2 + 2\text{NaOH}$ .

### 8.2.5.30 Index

The **Index** structure type shall be used to group the content, and the sub-structure representing the full semantics for that content, which is intended to be consumed as an index. A document may include multiple indexes representing different types of index, in which case, each index shall be tagged as a separate index. For each index, there should be a single **Index** structure element enclosing all content and sub-structure comprising that content.

Within a single index item, the **Reference** structure element should enclose the content and any interactive link that targets a location for the topic within the document. When an index item refers to multiple different locations within the document, each reference should be enclosed in a separate **Reference** structure element.

EXAMPLE If an index is an ordered list of alphabetically ordered references, then that list is structured with an **L** structure element within an **Index** structure element, the ordering is specified explicitly in the **ListNumbering** attribute and in the **LBody** structure elements, **Reference** structure elements are used to specify the internal references.

For each **Reference** structure element indicating a range of content within an **Index** structure element, there should be a corresponding **Ref** entry within it identifying each structure element enclosing that referenced content.

### 8.2.5.31 Bibliographic entry (BibEntry)

Content within a document that represents a bibliography entry shall be enclosed in a **BibEntry** structure element. For such bibliography entries, there should be a **Ref** entry in their respective structure element dictionaries that identifies all the places (if any) within the document that refer to that specific entry.

For a section of the document that is intended to be a bibliography, an ARIA (Accessible Rich Internet Application) attribute (see 8.2.6.4) in accordance with WAI-ARIA Module 1.0 shall be provided for the corresponding **Sect** structure element, indicating that the section type is *doc-bibliography*.

EXAMPLE A use of ARIA roles to identify a section intended to be consumed as a bibliography.

```
3 0 obj                                %Structure element for /Sect
<<
  /Type /StructElem
  /S /Sect                             %Structure type
  /A <<
    /O /ARIA-1.1                      %ARIA attribute owner
    /role /doc-bibliography           %DPUB role "doc-bibliography"
  >>
  /NS ...                             %Indirect reference to the PDF 2.0
                                     %namespace
  /P ...                             %Parent in structure hierarchy
  /K ...                             %Kids in the structure hierarchy
>>
endobj
```

### 8.2.5.32 Code

For content within a document that represents a fragment of computer code, that content shall be enclosed in a **Code** structure element.

NOTE The author determines what is considered as code.

EXAMPLE An author can consider an ASN.1 type definition to be code.

Where the content of the **Code** structure element is intended to be consumed as text, a textual representation of that code shall be present. A **Code** structure element may have an **Alt** property describing the code.

### 8.2.5.33 Artifact

Use of the **Artifact** structure type shall conform with the requirements in [8.3](#).

## 8.2.6 Structure attributes

### 8.2.6.1 General

All structure attributes defined in ISO 32000-2:2020, 14.7.6, along with their attribute owners as defined in ISO 32000-2:2020, 14.8.5.2, and in [8.2.5.14.2](#), may be present in a PDF file that conforms to this document. These attributes may be used as appropriate on all elements, regardless of namespace. Custom attribute owners and/or namespaces may be used.

NOTE ISO 32000-2 adds new attributes and new attribute values.

### 8.2.6.2 Layout

Given that layout (position, colour, etc.) is not encoded into PDF content streams in a way that strongly associates stylistic properties with content, structure elements shall include layout attributes in accordance with ISO 32000-2:2020, 14.8.5.4 to fully convey the semantics of a given use of colour, contrast, format or layout as necessary, to reflect all intended meaning in a manner that addresses the broadest-possible range of consuming technologies.

NOTE 1 This document does not intend to require complete descriptions of page layouts. However, layout information can help to provide assistive technology users with information about the material they are reading.

Attributes shall be present when the relevant semantic property is present in the content, has semantic significance and differs from any default or inherited value. Any attribute may be present to the extent that the property exists in the content. An attribute should not be present if the property does not exist in the content.

NOTE 2 [Annex B](#) provides examples of semantic significance for the various attribute types. Where semantic significance is marginal or unlikely, no examples are provided.

### 8.2.6.3 Table, List, PrintField, Artifact

Standard structure types to which *Table* ([8.2.5.26](#)), *List* ([8.2.5.25](#)), *PrintField* ([8.10.4](#)) and *Artifact* ([8.3.1](#)) attributes apply shall use their respective attributes.

NOTE [Annex B](#) provides information on considering the semantic relevance of attributes.

### 8.2.6.4 ARIA

ARIA has utility in PDF by providing a set of roles which extend the semantic structures defined in HTML and PDF. In PDF, ARIA roles serve at least two purposes:

- expanding the semantics of a given structure element (e.g. identify content enclosed in an **FENote** structure element as a footnote or endnote);
- providing semantics for non-semantic elements (e.g. identify that a **Div** structure element encloses a complex infographic).

The *ARIA-1.1* attribute owner, introduced in ISO 32000-2:2020, 14.8.5.2, may be used to include ARIA (version 1.1) attributes on any structure element. ARIA attributes shall be semantically appropriate for the use cases defined in the WAI-ARIA Module 1.0.



ARIA (version 1.1) and DPUB-ARIA roles, if present, shall use the *ARIA-1.1* attribute owner. When a role is specified, it shall not contradict the semantic intent of the structure element to which it applies.

**EXAMPLE** The following code represents a possible use of ARIA roles to extend the semantics of the **Part** structure type to identify it as a glossary.

```
3 0 obj                                %Structure element
<</Type /StructElem
  /S /Part                             %Structure type
  /A <</O /ARIA-1.1                   %ARIA attribute owner
    /role /doc-glossary               %Digital Publishing role "doc-glossary"
  >>
  /NS 7 0 R                           %Indirect reference to the PDF 2.0
                                     %namespace
  /P ...                              %Parent in structure hierarchy
  /K ...                              %Kids in the structure hierarchy
>>
endobj
7 0 obj                                %Namespace dictionary
<</Type /Namespace
  /NS (http://iso.org/pdf2/ssn)       %PDF 2.0 namespace identifier
>>
endobj
```

ARIA offers other attributes beyond role, including those which can define the state and properties of a given structure element. States and properties may be used, but do not impact rendering of the PDF, nor its interactive features.

**NOTE** PDF does not have a live Document Object Model (DOM) in the way that HTML does. There is limited interaction within PDF, and PDF processors do not expect structure element attributes to dynamically change within the document. There is no means by which JavaScript can directly control the attributes on a structure element, which limits author control of these attributes to what is specified within the file.

## 8.3 Artifacts

### 8.3.1 General

Any content, except Annotations (see 8.9), that is not real content as specified in ISO 32000-2:2020, 14.8.2.2 shall be explicitly identified using one of the two methods defined in ISO 32000-2:2020, 14.8.2.2.2.

**NOTE 1** Rather than allow this identification to be implied by the content's absence from the structure tree, this requirement strengthens the recommendation to explicitly identify artifacts in ISO 32000-2:2020, 14.8.2.2.

Where an artifact property list entry, as defined in ISO 32000-2:2020, 14.8.2.2.2, is semantically appropriate for an artifact defined solely using a marked content sequence, they shall be present. Where artifact attributes, as defined in ISO 32000-2:2020, 14.8.5.8, are semantically appropriate for an **Artifact** structure element, they shall be present.

**NOTE 2** An artifact defined without an **Artifact** structure element uses property list entries associated with the marked content sequence. An artifact defined with an **Artifact** structure element uses structure attributes, not property list entries.

### 8.3.2 Use of the Artifact structure element

Where an artifact is only meaningful in the context of content in the structure tree, it shall be enclosed in an **Artifact** structure element.

**NOTE** The **Artifact** structure type is useful, for example, in the case of page content that is not intended for inline reading. One such case is a legal document in which each line is numbered for reference purposes. In this case, a **Sub** structure element in accordance with ISO 32000-2:2020, 14.8.4.6 can enclose each line within a **P** (paragraph) structure element. Within each **Sub**, the use of an **Artifact** structure element to enclose the line-number serves to distinguish the line-number for reading purposes while retaining the context for reference purposes. Use of an **Lbl** structure element in this case (as implied in ISO 32000-2:2020, 14.8.4.6) is inappropriate because the line-numbers (a mechanical artifact, akin to pagination) would thereby be included in the document content.

## 8.4 Text representation for content

### 8.4.1 General

Conformity with this document requires that text content be programmatically determinable.

### 8.4.2 Unicode mapping in tagged PDF

As no pre-defined meaning is associated with Unicode Private Use Area (PUA) values, such values in content streams shall be used only if no other valid Unicode value is available.

NOTE The preceding paragraph differs from ISO 32000-2:2020 only in that use of the Unicode PUA when published meanings are available is directly contrary to the general requirements specified in this document.

### 8.4.3 Replacements and alternatives for text

**ActualText** shall be used when it is necessary to convey an alternative set of Unicode codepoints than what is generated by a processor based on ISO 32000-2:2020, 9.10.

For content mapped to a code or codes in the Unicode PUA, an **ActualText** entry as specified in ISO 32000-2:2020, 14.9.4 should be present for this character or for the sequence of characters of which such character is a part. In all cases, where real content maps to Unicode PUA values, an **ActualText** or **Alt** entry shall be present.

NOTE **ActualText** and **Alt** entries cannot use the Unicode PUA (see 8.6).

### 8.4.4 Declaring natural language

The language of all content and text strings shall be specified.

NOTE 1 Where a file includes Unicode text strings with two or more languages requiring three-letter language codes, it is possible that the language of some text is not representable.

The default natural language for content and text strings shall be specified using the **Lang** entry, with a non-empty value, in the document catalog dictionary.

Changes in the natural language shall be declared using the methods described in ISO 32000-2:2020, 14.9.2. Changes in natural language inside text strings (e.g. inside alternate descriptions) shall be declared using a language identifier as described in ISO 32000-2:2020, 7.9.2.2.

When specifying language using one of the methods above, a **Lang** entry shall resolve to a specific language.

NOTE 2 The declaration of natural language in the document catalog dictionary does not imply a primary language for the document. It is the responsibility of processors to consider how documents containing multiple languages are represented to end users to avoid the perception that the **Lang** entry in the document catalog dictionary represents a primary or otherwise dominant language.

## 8.4.5 Fonts

### 8.4.5.1 General

The intent of the requirements in 8.4.5.1 to 8.4.5.7 is to ensure that the future rendering of the textual content of a conforming file matches, on a glyph-by-glyph basis, the static appearance of the file as originally created and, when possible, to allow the recovery of semantic properties for each character of the textual content. Unless a requirement specifically states that it shall only apply to text rendered by a conforming processor, it shall apply to any font including those used exclusively with text rendering mode 3.

NOTE A font referenced solely in text rendering mode 3 (ISO 32000-2:2020, 9.3.6) is not rendered and is thus exempt from the requirements that impact the visual representation of the glyphs of a font.

### 8.4.5.2 Font types

All fonts and font programs used in a conforming file, regardless of rendering mode usage, shall conform to the requirements specified in ISO 32000-2:2020, 9.6 and 9.7, as well as to the font specifications referenced by requirements.

NOTE This document does not prescribe how such conformance is determined.

Multiple master fonts shall be considered a special case of Type 1 fonts. Any requirement regarding Type 1 fonts shall also be required for multiple master fonts.

### 8.4.5.3 Composite fonts

#### 8.4.5.3.1 General

For any given composite (Type 0) font within a conforming file, the **CIDSystemInfo** entry in its CIDFont dictionary and its Encoding dictionary shall have the following relationship:

- a) If the **Encoding** key in the Type 0 font dictionary has a value of *Identity-H* or *Identity-V*, then any values **Registry**, **Ordering**, and **Supplement** keys may be used in the **CIDSystemInfo** dictionary of the CIDFont.
- b) Otherwise, the corresponding values of the **Registry** and **Ordering** keys in both **CIDSystemInfo** dictionaries shall be identical, and the value of the **Supplement** key in the **CIDSystemInfo** dictionary of the CIDFont shall be greater than or equal to the value of the **Supplement** key in the **CIDSystemInfo** dictionary of the CMap.

NOTE The requirement for the **Supplement** key ensures that the font includes glyphs for all CIDs (Character Identifiers) which can be referenced by the CMap.

#### 8.4.5.3.2 CIDFonts

All embedded Type 2 **CIDFonts** in the CIDFont dictionary shall contain a **CIDToGIDMap** entry that shall be a stream mapping from CIDs to glyph indices or the name *Identity*, as described in ISO 32000-2:2020, Table 115.

#### 8.4.5.4 CMaps

All CMaps used within a file that conforms to this document, except those listed in ISO 32000-2:2020, Table 116, shall be embedded in that file, as described in ISO 32000-2:2020, 9.7.5. For those CMaps that are embedded, the integer value of the **WMode** entry in the CMap dictionary shall be identical to the **WMode** value in the embedded CMap stream.

A **CMap** shall not reference any other **CMap** except those listed in ISO 32000-2:2020, Table 116.

### 8.4.5.5 Embedding

#### 8.4.5.5.1 General

The font programs for all fonts used for rendering within a conforming file, as determined by whether at least one of its glyphs is referenced from one or more content streams, shall be embedded within that file, as defined in ISO 32000-2:2020, 9.9.

NOTE 1 Embedding the font programs allows any conforming processor to correctly reproduce all glyphs in the way they were originally published without reference to external resources.

NOTE 2 As discussed in ISO 32000-2:2020, 9.3.6, text rendering mode 3 specifies that glyphs are not stroked, filled, or used as a clipping boundary. A font referenced for use solely in this mode is therefore not rendered and is thus exempt from the embedding requirement.

Only font programs that are legally embeddable in a file for unlimited, universal rendering shall be used.

NOTE 3 This document precludes the embedding of font programs whose legality depends upon special agreement with the copyright holder. This requirement can create an unacceptable burden to verify the existence, validity, and longevity of such claims.

Embedded fonts shall define all glyphs referenced for rendering within the conforming file.

NOTE 4 As stated in 8.4.5.5.2, subsets of a font are acceptable if the embedded font provides glyph definitions for all characters referenced within the file.

#### 8.4.5.5.2 Subset embedding

Conforming files may include embedded subsets of font programs in accordance with ISO 32000-2:2020, 9.6 and 9.7.

NOTE The use of subsets of a font and its associated font program allows a potentially substantial reduction in the size of conforming files.

#### 8.4.5.6 Font metrics

For every font embedded in a conforming file and referenced for rendering, the glyph width information in the font dictionary and in the embedded font program shall be consistent for every glyph. Glyphs that are referenced only with rendering mode 3 are exempt from this requirement.

If a composite (Type 0) font is used for rendering in vertical writing mode as specified in ISO 32000-2:2020, 9.7.4.3 and if the associated embedded font program contains information about vertical metrics of the glyphs, this information shall be consistent with the values of the **DW2** and **W2** keys in the font descriptor dictionary.

For this document, consistent is defined to be a difference of no more than 1/1 000 unit in text space, consistent with ISO 32000-2:2020, 9.2.4.

NOTE Conformance with this subclause ensures predictable font rendering, regardless of whether a given processor uses the metrics in the font dictionary or those in the font program.

#### 8.4.5.7 Character encodings

For all non-symbolic TrueType fonts used for rendering, the embedded TrueType font program shall contain at least the Microsoft Unicode (3, 1 – Platform ID = 3, Encoding ID = 1), or the Macintosh Roman (1, 0 – Platform ID = 1, Encoding ID = 0) “cmap” subtable.

NOTE Conformance with this requirement ensures that all necessary glyph lookups can be carried out.

All non-symbolic TrueType fonts shall have either *MacRomanEncoding* or *WinAnsiEncoding* as the value for the **Encoding** key in the Font dictionary, or as the value for the **BaseEncoding** key in the dictionary that is the value of the **Encoding** key in the Font dictionary.

In addition, all non-symbolic TrueType fonts shall not define a **Differences** array unless all the glyph names in the **Differences** array are listed in the Adobe Glyph List and the embedded font program contains at least the Microsoft Unicode (3, 1 – Platform ID = 3, Encoding ID = 1) encoding in the “cmap” subtable.

Symbolic TrueType fonts shall not contain an **Encoding** entry in the font dictionary. The “cmap” subtable in the embedded font program shall either contain the Microsoft Symbol (3, 0 – Platform ID = 3, Encoding ID = 0) or the Mac Roman (1, 0 – Platform ID = 1, Encoding ID = 1) encoding.

In all cases for TrueType fonts that are to be rendered, character codes shall be able to be mapped to glyphs in accordance with ISO 32000-2:2020, 9.6.5, without the use of a non-standard mapping chosen by the conforming processor.

#### 8.4.5.8 Unicode character maps

The font dictionary of all fonts, regardless of their rendering mode usage, shall include a **ToUnicode** entry whose value is a CMap stream object that maps character codes for at least all referenced glyphs to Unicode values, as described in ISO 32000-2:2020, 9.10.3, unless the font meets at least one of the following four conditions:

- it uses the predefined encodings *MacRomanEncoding*, *MacExpertEncoding* or *WinAnsiEncoding*, as defined in ISO 32000-2:2020, Annex D;
- it is a Type 1 or Type 3 font where the glyph names of the glyphs referenced are all contained in the Adobe Glyph List, or the set of named characters in the Symbol font, as defined in ISO 32000-2:2020, Annex D;
- it is a Type 0 font whose descendant CIDFont uses the Adobe-GB1, Adobe-CNS1, Adobe-Japan1 or Adobe-KR-9 character collections;
- it is a non-symbolic TrueType font.

NOTE Unicode mapping allows the retrieval of semantic properties regarding every character referenced in the file.

If a **ToUnicode** CMap is present, the Unicode values it specifies shall all be greater than zero (0), but not equal to either U+FEFF or U+FFFE.

#### 8.4.5.9 Use of .notdef glyph

A file in conformance with this document shall not contain a reference to the **.notdef** glyph as defined in ISO 32000-2:2020, 9.7.6.3, from any of the text showing operators, regardless of text rendering mode, in any content stream.

NOTE Since the **.notdef** glyph does not have any semantic value, this requirement is present to avoid any ambiguity which can result from its use.

### 8.5 Real content without textual semantics

#### 8.5.1 General

Real content that does not possess the semantics of text objects and does not have an alternate textual representation shall be enclosed within **Figure** structure elements in accordance with ISO 32000-2:2020, 14.8.4.8.5, or **Formula** structure elements in accordance with ISO 32000-2:2020, 14.8.4.8.6, as appropriate.

NOTE Although encoded as text, ASCII art typically possesses semantics appropriate to the **Figure** structure type. Likewise, although encoded (to take one possible example) as non-textual objects, in a conforming file, a set of paths can possess the semantics appropriate to structure elements typically used for text, so long as they include appropriate **Alt** and/or **ActualText** attributes.

#### 8.5.2 Use of ActualText

When real content is intended to be consumed as text, but is not present as text objects, and is therefore not inherently interpretable as text by a PDF processor, an **ActualText** property as specified in ISO 32000-2:2020, 14.9.4 shall be provided.

Structure elements that include an **ActualText** property shall be, or shall be enclosed within, a semantically appropriate structure element.

NOTE The definition of **ActualText** in ISO 32000-2:2020, Table 355 changed from the definition in ISO 32000-1:2008, Table 323.

EXAMPLE It is common for text to be represented from native PDF vector drawing operations, i.e. without text objects.

## 8.6 Text string objects

Text strings (see ISO 32000-2:2020, 7.9.2.2) intended to be human readable shall not use the Unicode PUA.

NOTE Text string type is a subtype of string type and represents data encoded using specific conventions. It is not used for content within content streams.

EXAMPLE Bookmarks (outlines) are an example of a user-facing text string.

## 8.7 Optional content

All optional content configuration dictionaries in the file, including the default optional content configuration dictionary, shall contain a **Name** entry (see ISO 32000-2:2020, Table 96) whose value is a non-empty text string when:

- a) a file contains a **Configs** entry in the **OCProperties** entry of the document catalog dictionary (see ISO 32000-2:2020, Table 29), and
- b) the **Configs** entry contains at least one optional content configuration dictionary.

The **AS** key shall not appear in any optional content configuration dictionary.

NOTE 1 This prevents the automatic adjustment of states based on usage information (see ISO 32000-2:2020, 8.11.4.5).

NOTE 2 The requirements of 8.4.5 apply for all fonts used in all optional content, even where a particular exchange does not result in some optional content being rendered.

## 8.8 Intra-document destinations

All destinations (see ISO 32000-2:2020, 12.3.2) whose target lies within the same document shall be structure destinations (see ISO 32000-2:2020, 12.3.2.3).

NOTE 1 This requirement applies to link annotations providing intra-document navigation.

Where content actionably points to other content, such as links occurring in the context of captions, tables of contents, cross references, hyperlinks, or other links within a document, such content shall use one or both of the following mechanisms:

- point to the other content by means of a link annotation using a Structure Destination (see ISO 32000-2:2020, 12.3.2.3);
- use the **Ref** entry (see ISO 32000-2:2020, Table 355) on the source structure element to reference the target structure element.

NOTE 2 Consuming software can choose to provide various navigation features leveraging the **Ref** entry to possibly provide richer navigation options than are available with link annotations alone.

NOTE 3 Footnotes and endnotes have specific requirements pertaining to the **Ref** entry (see 8.2.5.14).

## 8.9 Annotations

### 8.9.1 General

Annotations serve multiple purposes within PDF files, including (but not limited to):

- supporting document workflows, including review, approval and redaction;
- support for multimedia in PDF, including video, sound and 3D;
- support for interactive elements such as links and forms.



These distinct purposes mean that requirements vary for different annotation types and even potentially depending on the specific use of a given annotation type. This clause identifies requirements for each of these purposes except for forms (see [8.10](#)).

Annotation types deprecated in ISO 32000-2 shall not be present.

Annotations whose subtypes are not defined in ISO 32000-2 may be present and shall be in conformity with all requirements in [8.9](#).

## 8.9.2 Semantics and content

### 8.9.2.1 General

This clause describes semantic requirements for all annotation types defined in ISO 32000-2.

Annotations (see ISO 32000-2:2020, 12.5) shall be included in the structure tree unless explicitly excluded by the following subclauses.

When included in the structure tree, annotations shall use the most semantically appropriate structure type in accordance with [8.2.2](#).

The semantics of annotation types shall be defined by the enclosing parent structure element, or, where no semantically appropriate element exists, a respective **Annot** structure element.

NOTE 1 ISO 32000-2 permits use of the **Annot** structure type for all annotation types except Widget and Link, but **Annot** is not required for annotations.

The use of the **Annot** structure type does not itself imply any specific type of annotation.

NOTE 2 A single **Annot** structure element can enclose multiple annotation objects (see ISO 32000-2:2020, Table 368) when they are intended to be perceived as a single object. See [8.9.4](#) for requirements related to alternate descriptions in this context.

EXAMPLE A set of markup annotation objects, such as strikethrough, caret and highlight, can represent a single intent.

NOTE 3 Annotation objects cannot occur multiple times in the structure tree (see ISO 32000-2:2020, Table 166) because only one **StructParent** is allowed.

ISO 32000-2 enables substructure within annotation appearance streams via marked content references. Files in conformity with this document shall not use marked content references to substructure annotation appearance streams (see ISO 32000-2:2020, Table 357).

NOTE 4 The effect of the above clause is to require that annotations are included as whole objects in a single structure element.

[8.9.2.2](#) to [8.9.2.4](#) specify additional requirements specific to both classes of annotations and individual annotation types.

### 8.9.2.2 Annotations as artifacts

Any annotation, regardless of its type, may be an artifact (see [8.3](#)). Annotations that are artifacts shall either not be present in the structure tree or shall be enclosed within **Artifact** structure elements.

In addition, annotations meeting either of the following criteria shall be artifacts:

- The Invisible flag is set in accordance with ISO 32000-2:2020, Table 167;
- The NoView flag is set and the ToggleNoView flag is not set, in accordance with ISO 32000-2:2020, Table 167.

NOTE The Hidden flag defined in ISO 32000-2:2020, Table 167 does not imply that the annotation is always an artifact, as that flag often changes during the user's workflow, for example with Widget annotations.

### 8.9.2.3 Markup annotations

Markup annotations defined in ISO 32000-2:2020, 12.5.6.2 and ISO 32000-2:2020, Table 171 include the following annotation types:

- Text (also known as sticky notes) (defined in ISO 32000-2:2020, 12.5.6.4)
- Free text (defined in ISO 32000-2:2020, 12.5.6.6)
- Line (defined in ISO 32000-2:2020, 12.5.6.7)
- Square and circle (defined in ISO 32000-2:2020, 12.5.6.8)
- Polygon and polyline (defined in ISO 32000-2:2020, 12.5.6.9)
- Text markup (Highlight, Underline, Squiggly-underline, Strikeout) (defined in ISO 32000-2:2020, 12.5.6.10)
- Caret (defined in ISO 32000-2:2020, 12.5.6.11)
- Rubber stamp (defined in ISO 32000-2:2020, 12.5.6.12)
- Ink (defined in ISO 32000-2:2020, 12.5.6.13)
- File attachment (defined in ISO 32000-2:2020, 12.5.6.15)
- Redaction (defined in ISO 32000-2:2020, 12.5.6.23)

As annotation semantics are defined by annotation type, markup annotations shall be enclosed within **Annot** structure elements.

NOTE 1 Markup annotations are only intended to annotate real content, not to represent it.

Markup annotations may omit **RC** and **Contents** entries, but if the context of the annotation (other than a free text annotation) is insufficient to convey its purpose, a **Contents** entry shall be present to convey that purpose. When both **RC** and **Contents** entries are present, they shall be textually equivalent.

NOTE 2 Free text annotations use the **RC** and **Contents** entries to provide their displayed text.

NOTE 3 Sighted users often benefit from visual clues derived from the visual representation of annotations that other users are not able to perceive. **RC** and **Contents** entries can provide greater context to assist with consumption.

NOTE 4 Content in an **RC** entry has appearance characteristics, including colour contrast and other visual qualities, that require accessibility consideration (for example, colour-contrast in WCAG 2.2<sup>[5]</sup>).

### 8.9.2.4 Annotation types

#### 8.9.2.4.1 Text

Text annotations shall follow the requirements in [8.9.2.3](#).

The location of text annotations in the structure tree, combined with their respective **Contents** entries, should provide sufficient context to understand each annotation's intent.

NOTE [8.9.3.1](#) explains the rationale for this provision.

The value of the **Name** entry should match the author's semantic intent for the annotation.



#### 8.9.2.4.2 Link

Link annotations should include a **Contents** entry to enrich information available to assistive technology.

NOTE 1 Link annotations are often accessed out of context; the **Contents** entry provides optional additional information. The **Contents** entry is also particularly valuable in the context of link targets that are not intended to be human-readable.

EXAMPLE A link over the text “click here” is improved by a **Contents** entry to advise the user regarding the link’s target.

See [8.2.5.20](#) for requirements specific to the **Link** and **Reference** structure types.

See [8.13.3](#) for requirements specific to link annotations that use URI actions.

The content associated with a link annotation shall be contiguous in the logical reading order.

NOTE 2 This requirement implies that non-contiguous linked content always uses multiple link annotations.

#### 8.9.2.4.3 Free text

Free text annotations shall follow the requirements in [8.9.2.3](#).

If the free text annotation includes an intent of *FreeTextCallout*, in accordance with the **IT** key defined in ISO 32000-2:2020, Table 177, then the logical reading order should align with the location of the callout’s origination.

#### 8.9.2.4.4 Line, square, circle, polygon and polyline

Line, square, circle, polygon and polyline annotations shall follow the requirements of [8.9.2.3](#).

EXAMPLE For a line connecting two paragraphs, the respective **P** structure elements can include **Ref** entries that reference the **Annot** structure element that encloses the respective line annotation. Further, that **Annot** structure element can include a **Ref** entry that references the related **P** structure elements.

#### 8.9.2.4.5 Text markup

Text markup annotations shall follow the requirements of [8.9.2.3](#).

If a text markup (highlight, underline, squiggly or strikeout) annotation is associated with content that is non-contiguous in the logical reading order, separate text markup annotations should be used for each contiguous unit of content.

NOTE For a highlight spanning two pages, a single **Annot** structure element can include object references to the two highlight annotations.

#### 8.9.2.4.6 Caret

Caret annotations shall follow the requirements of [8.9.2.3](#).

#### 8.9.2.4.7 Rubber stamp

Rubber stamp annotations shall follow the requirements of [8.9.2.3](#).

If the **Name** entry, as defined in ISO 32000-2:2020, 12.5.6.12, is insufficient to describe the intent of the stamp annotation, a **Contents** entry describing the author’s intent shall be provided.

#### 8.9.2.4.8 Ink

Ink annotations shall follow the requirements of [8.9.2.3](#).

A **Contents** entry describing the author's intent shall be provided.

NOTE As Ink annotations are freeform, descriptive information is necessary for accessibility. For example, a handwritten signature for John Doe can be described as "the signature of John Doe".

#### 8.9.2.4.9 Popup

Popup annotations shall not be present in the structure tree.

NOTE Popup annotations do not define any semantics and are solely defined in the context of their parent markup annotation.

#### 8.9.2.4.10 File attachment

File attachment annotations shall follow the requirements of [8.9.2.3](#).

NOTE 1 A sufficient description can come from the **Desc** entry of a file specification dictionary referenced by the file attachment annotation. The **Contents** entry of the file attachment annotation takes precedence over the **Desc** entry if more context is required, in accordance with ISO 32000-2:2020, 12.5.6.15.

When a file attachment annotation (as defined in ISO 32000-2:2020, 12.5.6.15) references a file specification dictionary (as defined in ISO 32000-2:2020, 7.11.3), the file specification dictionary shall include an **AFRelationship** entry.

NOTE 2 File specifications can be a string or a dictionary, as defined in ISO 32000-2:2020, 7.11.1. Using the string form is equivalent to the **AFRelationship** entry having the value of *Unspecified*.

#### 8.9.2.4.11 Sound, movie

Sound and movie annotations are deprecated in PDF 2.0 and therefore, in accordance with [8.9.1](#), shall not be present.

NOTE The functions of sound and movie annotations have been replaced by multimedia features specified in ISO 32000-2:2020, Clause 13.

#### 8.9.2.4.12 Screen

Screen annotations shall include a **Contents** entry.

NOTE When including video or audio using screen annotations, in support of accessibility, third party accessibility requirements such as WCAG<sup>[5]</sup> typically apply.

#### 8.9.2.4.13 Widget

Widget annotations are intended to facilitate interactive forms, see [8.10](#).

In addition to the requirements of [8.9.2.2](#), a widget annotation of zero height and width shall be an artifact in accordance with [8.3](#).

NOTE 1 When used as part of a document workflow that makes use of signature profiles such as PAdES-B-LTA, widget annotations associated with document timestamp signatures are often artifacts in this sense. In such workflows, the number of timestamps that will be added over time can typically not be predicted in advance. Since modifying the structure tree in incremental updates after signing complicates signature validation, it is important that invisible widgets be exempt from any tagging requirements otherwise imposed by this document.

NOTE 2 Widget annotations that are artifacts, regardless of visibility, can nevertheless have their Print flag set [e.g. in cases where compliance with ISO 19005-4 (PDF/A-4) is required].

#### 8.9.2.4.14 Printer's mark

A printer's mark annotation shall be an artifact.

#### 8.9.2.4.15 Trap network

Trap network annotations are deprecated in PDF 2.0 and shall not be used in files conforming to this document, in accordance with [8.9.1](#).

#### 8.9.2.4.16 Watermark

When used as real content, Watermark annotations shall follow the requirements of [8.9.2.3](#).

NOTE For the purposes of this document, the general requirements for markup annotations also apply to Watermark annotations.

#### 8.9.2.4.17 Redaction

Redaction annotations shall follow the requirements of [8.9.2.3](#).

NOTE 1 The requirements of [8.9.2.2](#) and this subclause have no bearing on redactions themselves. They apply only to redaction annotations. On the other hand, the redaction artifact subtype specified in ISO 32000-2:2020, Table 363 and Table 385 does not apply to redaction annotations, but to redactions themselves (e.g. black boxes resulting from the application of redaction annotations to the document).

Where possible (considering the use of **QuadPoints** defined in ISO 32000-2:2020, Table 195), a single logical redaction shall be enclosed within a single Redaction annotation.

NOTE 2 This requirement implies that a single redaction spanning two or more lines on a single page would occur within a single Redaction annotation.

#### 8.9.2.4.18 Projection

Projection annotations shall follow the requirements in [8.9.2.3](#).

#### 8.9.2.4.19 3D and RichMedia

3D and RichMedia annotations shall include alternate descriptions in their respective **Contents** entries.

NOTE When including 3D, video or audio using 3D or RichMedia annotations, in support of accessibility, third party requirements such as WCAG [\[5\]](#) typically apply.

#### 8.9.2.4.20 Other

Annotations whose subtypes are not defined in ISO 32000-2 shall meet the requirements of this document.

### 8.9.3 Order

#### 8.9.3.1 General

Many users benefit from a visual association between annotations and real content. The placement of annotations in the structure tree, together with an understanding of the overlap between the annotations and associated content, ensures that annotations become part of the logical content order together with other real content in the document.

#### 8.9.3.2 Association with content

In the case of text markup, redaction and link annotations, the visual association between content and an annotation is represented by a **QuadPoints** entry or a **Rect** entry.

When included in the structure tree, annotations shall be placed as close to the annotated content as possible, as either:

- a child or sibling of the applicable structure element enclosing the real content, or

— a child of an **Annot**, **Link** or **Reference** structure element.

NOTE The content associated with a link annotation is defined by the Link or Reference structure element pointing to it. Any geometric association via **QuadPoints** or **Rect** is secondary.

To more closely associate annotations to the content they annotate, **Ref** entries may be used to associate structure elements enclosing those annotations with one or more structure elements enclosing that real content.

### 8.9.3.3 Tab order

Every page that includes an annotation shall contain a **Tabs** entry in its page dictionary in accordance with ISO 32000-2:2020, Table 31, and its value shall be *A*, *W* or *S*.

NOTE Regardless of the value of the **Tabs** entry, each annotation's location in the logical structure is defined by the structure tree.

## 8.9.4 Alternative descriptions

### 8.9.4.1 Contents entries

For many annotation types, the **Contents** entry serves as one mechanism for providing alternative descriptions. If an annotation requires an alternative description and if the **Contents** entry is available for that purpose, the alternative description shall be included using the **Contents** entry. However, for annotation types where the **Contents** entry is not intended as an alternative description, and if an annotation requires an alternative description, one of the other mechanisms defined in ISO 32000-2:2020, 14.9.3 shall be used.

### 8.9.4.2 Alt properties

Where an annotation has a **Contents** entry providing an alternative description and the directly enclosing structure element has an **Alt** entry, the values of **Alt** and **Contents** shall be identical.

## 8.10 Forms

### 8.10.1 General

Widget annotations (see ISO 32000-2:2020, 12.5.6.19), which comprise the interactive aspect of forms on the page, are referenced from the structure tree, whereas fields (see ISO 32000-2:2020, 12.7.4) provide the functional representation of forms.

Each widget annotation shall be enclosed by a **Form** structure element (see [8.2.5.22](#)) unless the widget annotation is an artifact.

EXAMPLE A widget annotation used exclusively for calculations is enclosed within an **Artifact** structure element.

A **Form** structure element shall enclose at most one widget annotation.

NOTE **Form** structure elements include individual widgets, several of which can together comprise a single field. As such, a **Form** structure element does not semantically equate to a form field.

A **Form** structure element may be used to enclose the content in non-interactive forms (forms without a widget annotation) (see [8.10.4](#)).

XFA forms (see ISO 32000-2:2020, Annex K) shall not be present.

## 8.10.2 Context

### 8.10.2.1 General

The context for a widget annotation shall be provided through a combination of the following:

- the real content surrounding the widget annotation;
- the position and grouping of structure elements containing the **Form** structure element (see [8.10.4](#));
- the field's label, if any (see [8.10.2.2](#));
- the field's **TU** entry, if any (see [8.10.2.4](#));
- the widget's label, if any (see [8.10.2.2](#));
- the widget's **Contents** entry, if any (see [8.10.2.3](#)).

NOTE The field's name (its **T** entry) does not contribute to conveying the field's context.

The context provided by these means shall provide users with sufficient information to understand an interactive form field and any associated widget annotations.

EXAMPLE 1 A set of radio buttons to choose ice-cream flavours labelled "Vanilla", "Strawberry", and "Chocolate" that can be fully described by their respective labels together with the **TU** entry on the associated field, would not require a **Contents** entry.

EXAMPLE 2 Each radio button in a survey form in which the column headers of a table contain the labels "Agree" and "Disagree" for the widgets in their respective columns, would require a **Contents** entry.

Where field types provide additional mechanisms for context, those mechanisms shall be considered.

### 8.10.2.2 Labels

Where real content is used to label a widget annotation, that real content shall be enclosed in one or several **Lbl** structure elements, which shall in turn be a direct descendent of a **Form** structure element that also includes the object reference to the widget annotation.

Where real content is used as the label for a group of widget annotations, that real content shall be enclosed in one or several **Lbl** structure elements. In turn, the **Lbl** structure element(s) shall be contained within the parent structure element that also contains, directly or indirectly, the **Form** structure element for each widget.

NOTE A label for a group of **Form** structure elements is considered a form field's label.

EXAMPLE 1 A form design application can allow an author to group questions comprising arbitrary text and/or structure elements such as lists, along with form controls, within a single semantic element.

EXAMPLE 2 A form design application can allow an author to group arbitrary content along with form controls.

### 8.10.2.3 Contents entry

If a label for a widget annotation is not present, or if the label is insufficient, a **Contents** entry shall be provided to supply description and context for the widget. The **Contents** entry should not be identical to the label because an identical entry does not add information.

NOTE ISO 32000-2:2020, Table 166 defines the **Contents** entry as providing alternative descriptions for some annotations, including widget annotations.

The field's **TU** entry may be used to complement or be identical to a widget's **Contents** entry.

EXAMPLE A cryptic label referencing information in nearby paragraphs or pages implies the need for a **Contents** entry.

If an additional action (**AA**) entry, as defined in ISO 32000-2:2020, Table 191, is present in a widget annotation dictionary, the respective widget's **Contents** entry shall describe the action's intent.

#### 8.10.2.4 TU entry

The **TU** entry in the field's dictionary is useful when there is higher-level context that a sighted user gets from their overview of the form. A **TU** entry may provide repetitive and useful information to facilitate easy consumption and may also duplicate higher-level content to maximize the reader's understanding of the field's context.

When a form field dictionary has multiple widget annotations, a **TU** entry is often insufficient to provide full context for each annotation, thus, a label, a **Contents** entry, or both, shall be present for each annotation to provide additional context.

NOTE Establishing context for AT users in complex forms presents many challenges. The **TU** entry provides a means of conveying additional or supplementary context.

### 8.10.3 Fields

#### 8.10.3.1 General

Fields may be read-only.

#### 8.10.3.2 Button fields

##### 8.10.3.2.1 General

For a widget annotation that is associated with a button field, the **Contents** entry, if present, shall reflect the intent of the **CA** entry in the widget's appearance characteristics dictionary (see ISO 32000-2:2020, Table 192).

NOTE The **CA** entry provides a default caption for the button which can be used to help understand the semantic purpose of the button.

##### 8.10.3.2.2 Push-button

For a widget annotation that is associated with a push-button field, the **Contents** entry, if present, shall also reflect the intent of the **RC**, **AC**, **IRI** and **IX** entries, in addition to the **CA** entry, in the widget's appearance characteristics dictionary (see ISO 32000-2:2020, Table 192).

NOTE These properties reflect the semantic intent of the user's interaction with the button.

If the **TU** entry is present in the field's dictionary, it shall reflect the intent of the **RC** entry in the widget's appearance characteristics dictionary.

##### 8.10.3.2.3 Check boxes and radio buttons

For checkboxes and radio buttons, both the field and its associated widgets shall conform to the requirements specified in [8.10.2](#).

EXAMPLE 1 A single check box button field containing multiple widget annotations representing individual checkboxes.

EXAMPLE 2 A single radio button field containing multiple widget annotations representing individual radio buttons.

NOTE Export values are intended for processing and are not intended to be descriptive.

### 8.10.3.3 Text fields

For text fields, when an **RV** entry (as defined in ISO 32000-2:2020, Table 228) is present, a **V** entry (as defined in ISO 32000-2:2020, Table 226) shall also be present, and they shall be textually equivalent.

### 8.10.3.4 Choice fields

For choice fields, the **Opt** entry specifies both the export value and the text for each choice as displayed to the user and, in this context, the text for each choice shall sufficiently convey its intent.

Both the field and its associated widget shall conform to [8.10.2](#).

### 8.10.3.5 Signature fields

If the location of a signature within a document has a bearing on its legal meaning, then the signature field's widget annotation, in accordance with [8.9.2.4.13](#), shall be treated as real content and shall be present in the structure tree. A signature field's widget annotation shall be considered an artifact if it meets the criteria defined in [8.9.2.2](#).

NOTE 1 As stated in ISO 32000-2:2020, 12.7.5.5, the location of a signature within a document can have a bearing on its legal meaning. The exact location of a signature relative to other content is essential to universal access.

NOTE 2 The Hidden flag has no impact on whether a given signature field's widget annotation is considered an artifact, see [8.9.2.2](#).

If a portion of the appearance of a signature is represented by a graphic, alternative text shall be provided for that graphic (see [8.9.4](#)).

The annotation's appearance shall not contradict the signature's metadata as found in the signature dictionary.

NOTE 3 Various metadata related to the signature such as date and time of signing, can be accessed through the signature field's signature dictionary. Accordingly, for accessibility purposes, processing the signature's appearance is not necessary.

## 8.10.4 Non-interactive forms

Non-interactive content representing a form field shall be enclosed within a **Form** structure element. Such a **Form** element shall include appropriate **PrintField** attributes as specified in ISO 32000-2:2020, 14.8.5.6.

NOTE 1 [B.3](#) provides examples for **PrintField** attributes.

NOTE 2 There is no requirement that any specific non-interactive content (e.g. database output formatted to look like an input form) represents a form field.

Where real content is used to label a non-interactive form field, that real content shall be enclosed in one or several **Lbl** structure elements, which shall in turn be a descendent of a **Form** structure element that also encloses the real content of the form field.

Where real content is used to label a group of non-interactive form fields, that real content shall be enclosed in one or several **Lbl** structure elements. In turn, the **Lbl** structure element(s) shall be contained within the parent structure element that also contains, directly or indirectly, the **Form** structure element for each form field.



## 8.11 Metadata

### 8.11.1 General

The **Metadata** stream as specified in ISO 32000-2:2020, 14.3 in the document catalog dictionary shall include a **dc:title** entry reflecting the title of the document.

NOTE 1 Consistent with deployment objectives, files can include an abstract, publisher information, publisher contact information, publication number, author, date, and copyright information in their XMP metadata.

NOTE 2 Other metadata can be present in files in conformity with this document.

### 8.11.2 Interactive aspects

The **ViewerPreferences** dictionary of the document catalog dictionary shall be present and shall contain at least the **DisplayDocTitle** key with a value of *true*, as defined in ISO 32000-2:2020, Table 147.

NOTE This makes the recommendation for **DisplayDocTitle** in ISO 32000-2:2020, Table 147 into a requirement.

## 8.12 Navigation

### 8.12.1 General

Navigation features facilitate effective orientation and efficient reading, especially on longer or highly structured documents.

### 8.12.2 Outlines

Longer documents should include a document outline as specified in ISO 32000-2:2020, 12.3.3. Each entry in an outline dictionary should include an Action as specified in [8.13](#).

NOTE 1 A document outline contains items that are often referred to as bookmarks. Outline items can refer to destinations or invoke actions.

NOTE 2 This document deliberately avoids defining any specific length as “longer”.

### 8.12.3 Page labels

If a page label as specified in ISO 32000-2:2020, 12.4.2 is present, it shall represent the same number as that perceived by a user.

When a page's number is not equal to one plus the page's index, page labels shall be present.

### 8.12.4 Article threads

Article threads as specified in ISO 32000-2:2020, 12.4.3, if present, should reflect the logical content order.

NOTE Article threads are not considered a content reuse or accessibility technology.

## 8.13 Actions

### 8.13.1 General

Conforming files may include actions as specified in ISO 32000-2:2020, 12.6.

### 8.13.2 ECMAScript actions

ECMAScript, if used, should enable processors to communicate changes to content, visibility, focus and location in an accessible manner.