SLOVENSKI STANDARD

SIST ISO 16684-1:2020

01-april-2020

Grafična tehnologija - Specifikacija razširljive metapodatkovne platforme (XMP) - 1. del: Podatkovni model, serializacija in glavne lastnosti

Graphic technology - Extensible metadata platform (XMP) specification - Part 1: Data model, serialization and core properties

Technologie graphique - Spécification de la plate-forme de métadonnées extensibles (XMP) - Partie 1: Modèle de données, mise en série et paramètres principaux

Ta slovenski standard je istoveten z: ISO 16684-1:2012

ICS:

35.240.30	Uporabniške rešitve IT v informatiki, dokumentiranju in založništvu	IT applications in information, documentation and publishing
37.100.99	Drugi standardi v zvezi z grafično tehnologijo	Other standards related to graphic technology

SIST ISO 16684-1:2020 **en**

INTERNATIONAL STANDARD

ISO 16684-1

First edition
2012-02-15

Graphic technology — Extensible metadata platform (XMP) specification —

Part 1: Data model, serialization and core properties

Technologie graphique — Spécification de la plate-forme de métadonnées extensibles (XMP) —
Partie 1: Modèle de données, mise en série et paramètres principaux

Reference number
ISO 16684-1:2012(E)

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

ISO 16684-1 was prepared by Adobe (as *XMP Specification Part 1, Data Model Serialization, and Core Properties*, July 2010) and was adopted, under a special “fast-track procedure”, by Technical Committee ISO/TC 130, *Graphic technology*, in parallel with its approval by the ISO member bodies.

ISO 16684 consists of the following parts, under the general title *Graphic technology — Extensible metadata platform (XMP) specification*:

— *Part 1: Data model, serialization and core properties*

Future parts will address formal validation of XMP and XML syntax for describing XMP UI elements.

Introduction

This International Standard specifies a standard for the definition, creation, and processing of metadata that can be applied to a broad range of resource types. The Extensible Metadata Platform (XMP) was introduced by Adobe Systems Incorporated in 2001 and has since established itself as a critical technology for improving business efficiency in many industries. The Adobe Systems XMP Specification Part 1 version of July 2010 is the basis for this International Standard. Establishing this International Standard ensures the stability and longevity of its definitions and encourages broader integration and interoperability of XMP with existing standards.

Metadata is data that describes the characteristics or *properties* of a resource. It can be distinguished from the main content of a resource. For example, for a word processing document, the *content* includes the actual text data and formatting information, while the *metadata* might include properties such as author, modification date, or copyright status.

Some information could be treated as either content or metadata, depending on context. In general, metadata is useful without regard for a resource's content. For example, a list of all fonts used in a document could be useful metadata, while information about the specific font used for a specific paragraph on a page would be logically treated as content.

Metadata allows users and applications to work more effectively with resources. Applications can make use of metadata, even if they cannot understand the native format of the resource's content. Metadata can greatly increase the utility of resources in collaborative production workflows. For example, an image file might contain metadata such as its working title, description, and intellectual property rights. Accessing the metadata makes it easier to perform such tasks as searching for images, locating image captions, or determining the copyright clearance to use an image.

File systems have typically provided metadata such as file modification dates and sizes. Other metadata can be provided by other applications, or by users. Metadata might or might not be stored as part of the resource with which it is associated.

This International Standard provides a thorough understanding of the XMP data model. It is useful for anyone who wishes to use XMP metadata, including both developers and end-users of applications that handle metadata for resources of any kind.

The serialization information is vital for developers of applications that will generate, process, or manage files containing XMP metadata. The serialization information will also interest application developers wishing to understand file content. This International Standard also provides additional guidelines for programmers who will implement XMP metadata processors.

The International Organization for Standardization (ISO) draws attention to the fact that it is claimed that compliance with this document may involve the use of a patent concerning the creation, processing, modification, and storage of XMP metadata.

ISO takes no position concerning the evidence, validity and scope of this patent right. The holder of this patent right has assured ISO that he is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO. Information may be obtained from:

Adobe Systems Incorporated
345 Park Avenue
San Jose, California, 95110-2704
USA

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. ISO shall not be held responsible for identifying any or all such patent rights.

Graphic technology — Extensible metadata platform (XMP) specification —

Part 1:
Data model, serialization and core properties

1 Scope

This part of ISO 16684 defines two essential components of XMP metadata:

- *Data model*: The data model is the most fundamental aspect. This is an abstract model that defines the forms of XMP metadata items, essentially the structure of statements that XMP can make about resources.
- *Serialization*: The serialization of XMP defines how any instance of the XMP data model can be recorded as XML.

In addition, this part of ISO 16684 defines a collection of *core properties*, which are XMP metadata items that can be applied across a broad range of file formats and domains of usage.

The embedding of XMP packets in specific file formats and domain-specific XMP properties are beyond the scope of this part of ISO 16684.

2 Normative references

The following referenced documents are indispensable for the application 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.

IEEE 754, *Standard for Binary Floating-Point Arithmetic*
http://grouper.ieee.org/groups/754/

IETF RFC 2046, *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*, November 1996
http://www.ietf.org/rfc/rfc2046.txt

IETF RFC 3066, *Tags for the Identification of Languages*, January 2001
http://www.ietf.org/rfc/rfc3066.txt

IETF RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*, January 2005
http://www.ietf.org/rfc/rfc3986.txt

Date and Time Formats, W3C submission, September 1997
http://www.w3.org/TR/NOTE-datetime

Dublin Core Metadata Element Set, Version 1.1, October 2010
http://dublincore.org/documents/dces/

Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C Recommendation 26 November 2008
http://www.w3.org/TR/2008/REC-xml-20081126/

Namespaces in XML 1.0 (Second Edition), August 2006
http://www.w3.org/TR/2006/REC-xml-names-20060816/

RDF/XML Syntax Specification (Revised), W3C Recommendation 10 February 2004
http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/

The Unicode Standard
http://www.unicode.org/standard/standard.html

URIs, URLs, and URNs: Clarifications and Recommendations 1.0, W3C Note 21 September 2001
http://www.w3.org/TR/2001/NOTE-uri-clarification-20010921/

3 Terms and definitions

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

3.1
character data
XML text that is not markup

[Extensible Markup Language specification, Section 2.4]

3.2
element content
XML text between the start-tag and end-tag of an element

[Extensible Markup Language specification, Section 3.1, syntax production 43]

3.3
empty-element tag
XML tag identifying an element with no content

[Extensible Markup Language specification, Section 3.1]

3.4
NCName
XML name that does not contain a colon (':', U+003A)

[Namespaces in XML, Section 3, syntax production 4]

3.5
property
named container for a metadata value at the top level of an XMP packet

NOTE Lower-level components of an XMP packet are structure fields, array items, and qualifiers.

3.6
RDF
Resource Description Framework, an XML syntax for describing metadata

[RDF/XML Syntax Specification]

3.7
rendition (of a resource)
resource that is a rendering of some other resource in a particular form

NOTE Various renditions of a resource have the same content in differing forms. For example, a digital image could have high resolution, low resolution, or thumbnail renditions. A text document could be in a word processor format for editing or rendered as a PDF for sharing. See also version (of a resource).

3.8
URI
Uniform Resource Identifier, a compact sequence of characters that identifies an abstract or physical resource

[IETF RFC 3986]

3.9
version (of a resource)
resource that is the result of editing some other resource

NOTE Different versions of a resource typically have differing content in the same form. See also rendition (of a resource).

3.10
XML element
primary component of XML syntax

[Extensible Markup Language specification, Section 3, syntax production 39]

3.11
XML expanded name
pair of strings consisting of a namespace URI and a local name

[Namespaces in XML, Section 2.1]

3.12
XMP processor
hardware or software component that is responsible for reading, modifying, or writing XMP

3.13
white space
XML text consisting of one or more space characters, carriage returns, line feeds, or tabs

[Extensible Markup Language specification, Section 2.3]

4 Notations

The following type styles are used for specific types of text:

Table 1 — Conventions for type styles

Typeface style	Used for
Bold	XMP property names. For example, **xmp:CreateDate**
Italic	Terms when defined in text, document titles, or emphasis.

The following names are used for important Unicode characters:

- SPACE - U+0020
- QUOTE - U+0022 (")
- APOSTROPHE - U+0027 (')

5 Conformance

5.1 General

Conforming XMP packets shall adhere to all requirements of this International Standard and conforming XMP packets are not required to use any feature other than those explicitly required by this International Standard.

NOTE The proper mechanism by which XML can presumptively identify itself as being an XMP packet is described in 7.3, “Optional outer XML”, and 7.4, “rdf:RDF and rdf:Description elements”.

5.2 Conforming readers

A conforming reader shall comply with all requirements regarding reader functional behaviour specified in this International Standard. The requirements of this International Standard with respect to reader behaviour are stated in terms of general functional requirements applicable to all conforming readers. A conforming reader shall accept all output from conforming writers, including optional output that conforming writers may produce. This International Standard does not prescribe any specific technical design, user interface, or implementation details for conforming readers.

5.3 Conforming writers

A conforming writer shall comply with all requirements regarding writer functional behaviour specified in this International Standard. The requirements of this International Standard with respect to writer behaviour are stated in terms of general functional requirements applicable to all conforming writers and focus on the creation of conforming XMP packets. This International Standard does not prescribe any specific technical design, user interface, or implementation details for conforming writers.

5.4 Conforming products

A conforming product shall comply with all requirements regarding reader and writer functional behaviour as specified in this International Standard.

6 Data model

6.1 XMP packets

An instance of the XMP data model is called an *XMP packet*. An XMP packet is a set of XMP metadata properties. Each property has a name and a value. Each property name in an XMP packet shall be unique within that packet.

NOTE 1 The restriction for unique names means that it is invalid to have multiple occurrences of the same property name in an XMP packet. Multiple values are represented using an XMP array value (see 6.3.4, “Array values”). Instead of having three **dc:subject** properties that each hold one keyword, there would be one **dc:subject** property that is an array with three items.

All properties in a single XMP packet shall describe a single resource. Separate XMP packets may describe the same resource. Conflict resolution for separate packets that describe the same resource is beyond the scope of this International Standard.

Lower-level components of an XMP packet (structure fields or array items) may describe one or more other resources.

NOTE 2 The provision for lower-level components about some other resource is not an addition to the data model, in that this is not a formal feature of the data model and is not reflected in written XMP in any specific manner. Rather, it is a clarification to the “one packet about one resource” rule, to avoid disallowing certain data models. The XMP about a compound resource might have a list of constituent resources and even copies of XMP about those constituents. This would all be modelled using the defined XMP value forms.

The composition of a resource and the precise association of an XMP packet with a resource is beyond the scope of this International Standard. Where feasible, an XMP packet should be physically associated with the resource that it describes.

NOTE 3 A common resource is a complete digital file, or an identifiable part of a digital file such as an embedded image in PDF. The structure of a PDF file and the manner of associating XMP with any particular component of a PDF file is beyond the scope of this part of ISO 16684.

The XMP packet that describes a digital file or part of a digital file should be embedded in the file using standard features of the file format to provide the association between the XMP packet and the resource. The embedding mechanisms for specific file formats are beyond the scope of this International Standard.

An XMP packet may contain a URI, called the *AboutURI*, that identifies the resource that the packet describes. The URI scheme, detailed URI syntax, and association of the URI with any target entity is beyond the scope of this International Standard.

NOTE 4 It is possible for an XMP packet to not contain an AboutURI and not have a physical association with the resource. Instead, there can be an external means of association.

EXAMPLE Consider the statement, “The author of *Moby Dick* is Herman Melville”. This statement is represented by metadata in which the resource is the book “Moby Dick”, the property name is “author”, and the property value is “Herman Melville”, as in Figure 1. (This is only a diagram, not an example of well-formed XMP.)

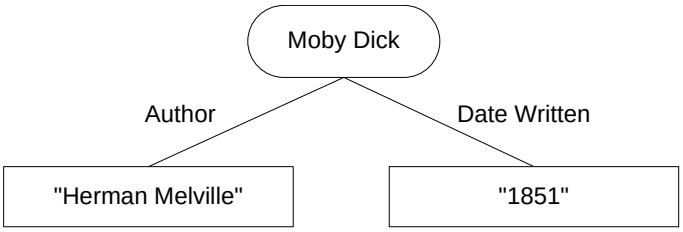

Figure 1 — Simple properties example diagram

NOTE 5 Notation such as that in Figure 1 is used in this International Standard to illustrate the XMP data model.

An XMP processor should accept all well-formed XMP as input, regardless of the data model expressed, and should by default preserve all unanticipated XMP when modifying a resource.

NOTE 6 The intent of these rules is that XMP is generally open to arbitrary extension of properties. Users of XMP are allowed to freely invent custom metadata and to expect XMP-aware applications to support the creation, modification, and viewing of that metadata. Therefore, this is expressed as a recommendation instead of as a requirement because any particular environment could have local policies about XMP usage.

6.2 XMP names

Properties (6.1, “XMP packets”) have names, as do fields of structure values (6.3.3, “Structure values”) and qualifiers (6.4, “Qualifiers”). All names in XMP shall be XML expanded names, consisting of a namespace URI and a local name. The namespace URI for an XMP name shall not be empty. Two XMP names shall be equivalent if their namespace URIs are identical and their local names are identical. This comparison shall be physical, byte-for-byte equality using the same Unicode encoding. Other processing, including but not limited to Unicode character normalizations, shall not be applied.

NOTE 1 XML namespace URIs are generally best viewed as string literals. There is no commitment that the URI is resolvable to a Web resource. Although many XML namespace URIs begin with "http://", there might be no HTTP page at that address.

The namespace prefix used in written XML—and, as a consequence, in XMP—serves only as a key to look up the appropriate URI. For convenience in this International Standard, XMP names are commonly written in a **prefix:local** style, for example, **dc:title**. The relevant URI for the prefix used in this document is either explicit, clear from local context, or irrelevant (as in the generic value-form diagrams where the specific URI does not matter).

NOTE 2 The specific convenience is that **dc:title** is more concise and readable than something like ("http://purl.org/dc/elements/1.1/", creator) in the cases where the namespace URI is known and meaningful. This is especially so when the precise URI is not relevant, as in an artificial example.

A namespace URI used in XMP should end in a character that is not allowed in an XML NCName (the local name). Recommended characters are the slash ("/", U+002F) or the number sign ("#", U+0023). This can improve compatibility with applications that concatenate the namespace URI and local name, avoiding potential collisions.

NOTE 3 The textual concatenation of a namespace URI and local name is seen in generic RDF processors that utilize the RDF triple notation. See B.3, "Namespace URI termination", for details.

Other than **xml:** and **rdf:**, all namespaces used in 6, "Data model", and Figure 5, "Qualifiers example", are illustrative. In particular, the "http://ns.adobe.com/xmp-example/" URI is fictional. The use of specific XMP names in the illustrations does not imply that they are defined in this International Standard. The namespaces defined in 8, "Core properties", are normative.

Following typical XML and World Wide Web practice, the creation of XMP names should use a namespace URI that incorporates a domain name owned by the creator. This diminishes the chance of namespace collisions and identifies the origin of the namespace.

In this International Standard, the **xml:** prefix is bound to the URI "http://www.w3.org/XML/1998/" that is defined in the Extensible Markup Language specification. The **rdf:** prefix is bound to the URI "http://www.w3.org/1999/02/22-rdf-syntax-ns#" that is defined in the RDF/XML Syntax Specification. The Extensible Markup Language specification and the RDF/XML Syntax Specification heavily restrict the use of these namespaces. Except for **rdf:type**, these namespaces shall not be used for any XMP property or structure field. Except for **rdf:type** and **xml:lang**, these namespaces shall not be used for any XMP qualifier. See also 7.9.2.5, "RDF Typed Nodes".

6.3 XMP value forms

6.3.1 General

Values in the XMP data model have one of three forms: simple, structure, or array. There are two variants of simple values: normal and URI. There are three variants of the array form: unordered array, ordered array, and alternative array. The fields in structures and the items in arrays may have any value form. There is no fixed bound on the complexity of XMP data modelling.

These forms are the primitive values of XMP. Higher-level data types may be defined that combine these primitive forms with additional constraints, such as those defined in 8, "Core properties".

6.3.2 Simple values

A simple value is a string of Unicode text as defined in The Unicode Standard. The string may be empty.

There are two variants of simple values: normal and URI. The URI variant of a simple value should be used for values that represent URIs; the normal variant should be used for all other simple values.

NOTE The distinction between normal and URI simple values is not critical to the organization of the abstract XMP data model. The distinction does have an effect on the RDF serialization, as seen in 7.5, "Simple valued XMP properties". This allows XMP data modelling to more closely align with general RDF data modelling.

EXAMPLE In Figure 2, the document XMP_Specification.pdf is shown with two properties, each with a simple value:

The value of the property **dc:format** is "application/pdf".

The value of the property **xmp:CreateDate** is "2002-08-15T17:10:04-06:00".

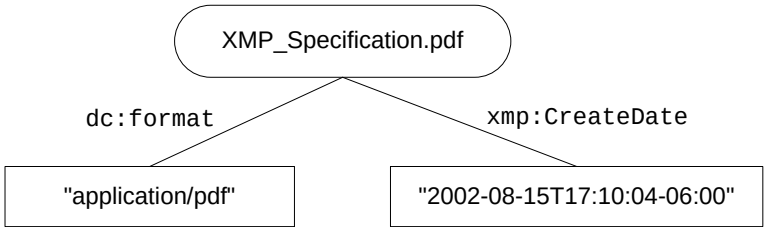

Figure 2 — Simple values example

6.3.3 Structure values

A structure is a container for zero or more named fields. The order of fields in a structure shall not be significant. Fields may be optional or required.

Each field in a structure shall have a unique name within that structure. Field names shall be XML expanded names. Fields need not be in the same namespace as their parent structure nor in the same namespace as other fields in the structure.

Each field in a structure may have any value form. The usage and consistency of fields in a given structure type is beyond the scope of this International Standard.

EXAMPLE Figure 3 shows a single structured property with three fields: **stDim:w** (width), **stDim:h** (height) and **stDim:unit** (units), whose values are "8.5", "11.0", and "inch".

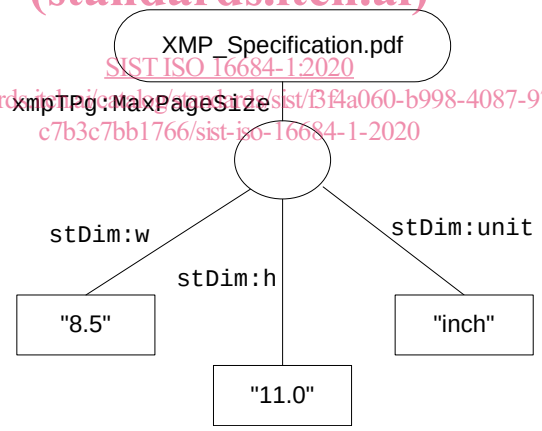

Figure 3 — Example of structure values

6.3.4 Array values

An array is a container for zero or more items indexed by ordinal position, starting from 1. The form of the array items may be any XMP value form. All items in an array shall have the same data type.

There are three variants of array: ordered, unordered, and alternative. The variant indicates the anticipated use of the array and constrains what XMP processors may do with it:

- An *unordered array* shall have no meaning or constraints on the order of items within it. The items in an unordered array may be reordered at any time.
- The items in an *ordered array* are ordered by their indices. The items in an ordered array shall not be arbitrarily reordered. The meaning of the order may be defined by data type or by application. Except for the data types defined in 8, “Core properties”, this International Standard does not specify any assumed or