INTERNATIONAL STANDARD

ISO 16684-1

Second edition
2019-04

Graphic technology — Extensible metadata platform (XMP) —

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 16684-1:2019(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: 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.

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 documents 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).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

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.

This document was prepared by Technical Committee ISO/TC 130, *Graphic technology*.

This second edition cancels and replaces the first edition (ISO 16684-1:2012), which has been technically revised.

A list of all parts in the ISO 16684 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.

Introduction

This document 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 document. Establishing this document 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 document 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 document 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 conformity 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) —

Part 1: Data model, serialization and core properties

1 Scope

This document 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 document 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 document.

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.

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

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*

SET D.C.M.E. Version 1.1, Octor 2010 http://dublincore.org/documents/dces/

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.

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

ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.1
character data
XML text that is not markup

[SOURCE: Extensible Markup Language specification, Section 2.4]

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

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

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

[SOURCE: Extensible Markup Language specification, Section 3.1]

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

[SOURCE: 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 1 to entry: Lower-level components of an XMP packet are structure fields, array items, and qualifiers.

3.6
RDF
Resource Description Framework
XML syntax for describing metadata

[SOURCE: RDF/XML Syntax Specification]

3.7
rendition
<resource>resource that is a rendering of some other resource in a particular form

Note 1 to entry: 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
compact sequence of characters that identifies an abstract or physical resource

[SOURCE: IETF RFC 3986]

3.9
version
<resource> resource that is the result of editing some other resource

Note 1 to entry: 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

[SOURCE: 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

[SOURCE: 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

[SOURCE: Extensible Markup Language specification, Section 2.3]

4 Notations

The following typeface 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 document and conforming XMP packets are not required to use any feature other than those explicitly required by this document.

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 adhere to all requirements regarding reader functional behaviour specified in this document. The requirements of this document 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 document does not prescribe any specific technical design, user interface, or implementation details for conforming readers.

5.3 Conforming writers

A conforming writer shall adhere to all requirements regarding writer functional behaviour specified in this document. The requirements of this document 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 document does not prescribe any specific technical design, user interface, or implementation details for conforming writers.

5.4 Conforming products

A conforming product shall adhere to all requirements regarding reader and writer functional behaviour as specified in this document.

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, which is an XML expanded name, and a value. Each such property name shall be unique within its direct node element, i.e. the highest level **rdf:Description** element to which it belongs.

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 document.

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 document. 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 document.

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 document.

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 document.

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.).

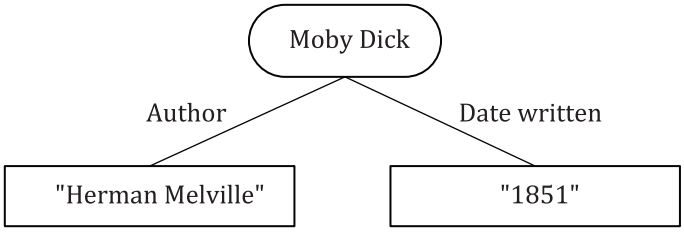

Figure 1 — Simple properties example diagram

NOTE 5 Notation such as that in Figure 1 is used in this document 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. Although many XML namespace URIs begin with “http://" there is no recommendation or requirement that the URI points to a web resource.

The namespace prefix used in XML — and, as a consequence, in XMP — serves only as a key to look up the appropriate URI. For convenience in this document, 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 Clause 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 document. The namespaces defined in Clause 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 document, 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 Clause 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".

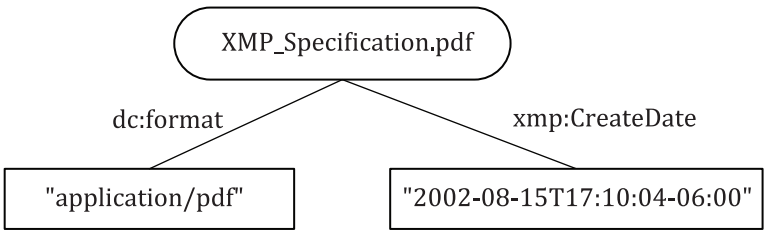

Figure 2 — Simple values example

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 document.

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".

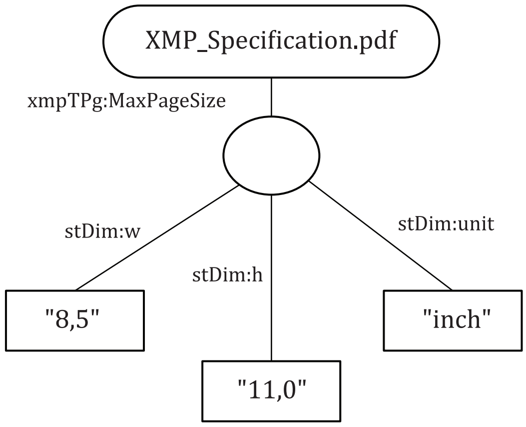

Figure 3 — Example of structure values

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 Clause 8, “Core properties”, this document does not specify any assumed or default meaning to the order of items in an ordered array.

— The items in an alternative array are ordered and 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 Clause 8, “Core properties”, this document does not specify any assumed or default ordering. If any item is a preferred default, it should be the first item in the array. The first item in the array should be chosen when no other criteria apply. An alternative array need not have an explicitly designed default item.

NOTE 1 The intent is that a reader who has no idea how to choose an item from the alternative array is encouraged to pick the first item.

NOTE 2 The anticipated usage of an unordered or ordered array is to consider all items together, such as an unordered list of keywords or an ordered list of authors. The anticipated usage of an alternative array is to select one item based on some criteria, for example, having multiple descriptions of a resource in various languages, then selecting one based on the user’s preferred language. Both ordered and alternative arrays have ordered items; the anticipated usage determines which array variant to use.

EXAMPLE Figure 4 shows an example of the Dublin Core property **dc:subject** (see 8.3, “Dublin Core namespace”), which is an unordered array. In this example, it contains three items.

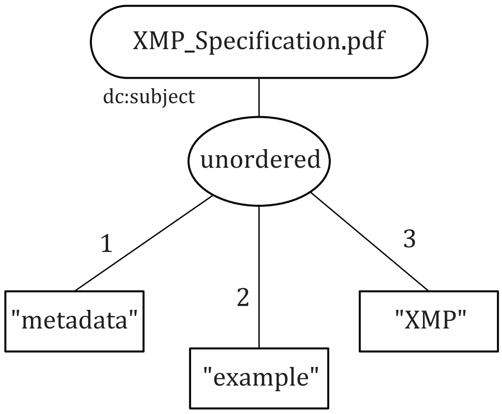

Figure 4 — Array values exampleF

6.4 Qualifiers

XMP qualifiers may be used to attach annotations to any XMP value, without changing the form of that value. For example, a simple value remains a simple value even when some XMP processor attaches arbitrary qualifiers to it. Qualifiers are metadata about the value to which they are attached. Each qualifier has a name and a value. The names shall be XML expanded names. The names of all qualifiers attached to a particular value shall be unique in that value. The value of a qualifier may be any XMP value form. A qualifier value may have qualifiers.

The **xml:lang** qualifier shall have a simple non-URI value and shall not have qualifiers on its value. An **xml:lang** qualifier on a structure or array should be considered a default language for the structure fields or array items. In accordance with IETF RFC 3066, the value of the **xml:lang** qualifier shall be a language code and all comparisons of **xml:lang** values shall be case-insensitive.

EXAMPLE Figure 5 shows an example of qualifiers.

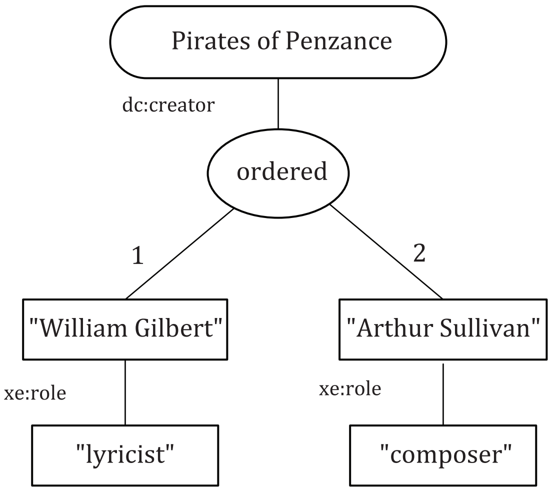

Figure 5 — Qualifiers example

7 Serialization

7.1 General

The abstract XMP data model needs a concrete representation when given a physical representation such as a digital file or a printed document. This document defines a canonical serialization of XMP

metadata using a subset of the RDF metadata syntax. The RDF serialization shall use Unicode text as defined in The Unicode Standard. The choice of Unicode encoding (UTF-8, UTF-16, or UTF-32) is beyond the scope of this document. Other file embedding or usage standards may specify the Unicode encoding.

For this serialization, a single XMP packet shall be serialized using a single **rdf:RDF** XML element.

Serialized XMP shall be well-formed XML and well-formed RDF. An XMP reader shall conform to the rules of XML and RDF given in their respective specifications.

An XMP reader shall recognize and honour a leading Unicode U+FEFF character as a byte-order marker. An XMP writer using UTF-16 or UTF-32 should include a leading Unicode U+FEFF character. An XMP writer using UTF-8 may include a leading Unicode U+FEFF character, although it is not recommended.

NOTE 1 One reason to avoid the U+FEFF with UTF-8 is that devices might exist that read only UTF-8 and are not prepared for a leading U+FEFF. The only rationale for using a leading U+FEFF with UTF-8 is as a clear encoding marker for when a reader might get either UTF-8 or UTF-16/32.

NOTE 2 The XMP serialization is intentionally presented as a fragment of an XML document, not as a fully formed XML document. That is, it is presented as a single outer XML element and element content, with no mention of the XML document prolog. This is done to allow the inclusion of multiple XMP packets in larger XML documents.

7.2 Equivalent RDF and XML

The normative statements in 7.4 to 7.8 define a canonical usage of the RDF syntax. Equivalent forms of RDF syntax that are allowed or prohibited are defined in 7.9, "Equivalent forms of RDF". Serialization of XMP uses only a subset of the RDF syntax. Parts of the RDF syntax not presented in this document shall not be used in serialized XMP.

Except where explicitly noted, XML white space, comments, and processing instructions may be written anywhere allowed by the RDF/XML Syntax Specification. Non-white character data is heavily constrained by RDF and XMP. It shall be used only in the element content of leaf elements that represent simple XMP values.

Comments and processing instructions may be ignored when reading and need not be preserved when updating an XMP packet.

NOTE 1 Phrases in this document such as "… element content shall consist of only …" do not constitute an explicit prohibition of white space, comments, or processing instructions. Such phrases restrict only the use of XML elements, attributes, and non-white character data.

NOTE 2 The purpose of the RDF serialization of XMP is to carry an instance of the XMP data model. XML comments and processing instructions have no effect on the XMP data model, no matter where they appear. White space outside the **rdf:RDF** element has no effect on the XMP data model. Allowed white space inside the **rdf:RDF** element has no effect on the XMP data model except when it is part of the element content for a simple value (7.5, "Simple valued XMP properties"), in which case it is part of the value.

All equivalent forms of XML text may be written. This includes but is not limited to:

Use of either an empty-element tag (of the form <ns:name/>) or an element with empty element content (of the form <ns:name></ns:name>).

— Use of either QUOTEs or APOSTROPHEs for attribute values.

— Order of attributes within an element.

— Distribution of xmlns attributes.

— The specific prefix associated with an XML namespace URI.