Adding Intelligence to Media

XMP SPECIFICATION PART 2

STANDARD SCHEMAS

Adobe

Extensible Metadata Platform (XMP) Specification: Part 2, Standard Schemas

Contents

Preface

This document set provides a complete specification for the Extensible Metadata Platform (XMP), which provides a standard format for the creation, processing, and interchange of metadata, for a wide variety of applications.

The specification has three parts:

- *Part 1, Data and Serialization Model*, covers the basic metadata representation model that is the foundation of the XMP standard format. The Data Model prescribes how XMP metadata can be organized; it is independent of file format or specific usage. The Serialization Model prescribes how the Data Model is represented in XML, specifically RDF.

  This document also provides details needed to implement a metadata manipulation system such as the XMP Toolkit (which is available from Adobe).

- *Part 2, Standard Schemas*, provides detailed property lists and descriptions for standard XMP metadata schemas; these include general-purpose schemas such as Dublin Core, and special-purpose schemas for Adobe applications such as Photoshop. It also provides information on extending existing schemas and creating new schemas.

- *Part 3, Storage in Files*, provides information about how serialized XMP metadata is packaged into XMP Packets and embedded in different file formats. It includes information about how XMP relates to and incorporates other metadata formats, and how to reconcile values that are represented in multiple metadata formats.

About this document

This document, *XMP Specification Part 2, Standard Schemas*, is intended for developers of applications that will generate, process, or manage files containing XMP metadata. Such developers may use either the XMP Toolkit provided by Adobe, or independent implementations.

Previously-defined formats (*native* formats) for still-image metadata, such as EXIF and IPTC/TIFF, represent information that is also represented by properties defined in standard XMP schemas. For information on how to reconcile property values among formats, and on how such reconciliation has been managed in Adobe applications, see *XMP Specification Part 3, Storage in Files*.

How this document is organized

This document has the following sections:

- Chapter 1, "XMP Schemas," explains how the schema definitions are presented, and provides details of property value types. It also describes how you can extend existing schemas or define new ones.

- Chapter 2, "XMP Standard Schemas," provides schema definitions for standard general-purpose schemas.

- Chapter 3, "Specialized Schemas," provides schema definitions for schemas that are specialized for Adobe applications or usages.

Conventions used in this document

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

Typeface Style	Used for:
`Monospaced bold`	XMP property names. For example, **`xmp:CreationDate`**
`Monospaced Regular`	XML code and other literal values, such as value types and names in other languages or formats

Where to go for more information

See these sites for information on the Internet standards and recommendations on which XMP Metadata is based:

Dublin Core Metadata Initiative	http://dublincore.org/
Extensible Markup Language (XML)	http://www.w3.org/XML/
IETF RFC 3066, Tags for the Identification of Languages	http://www.ietf.org/rfc/rfc3066.txt
ISO 639, Standard for Language Codes	http://www.loc.gov/standards/iso639-2/
ISO 3166, Standard for Country Codes	http://www.iso.ch/iso/en/prods-services/iso3166ma/index.html
IETF RFC 3986, Uniform Resource Identifier (URI): Generic Syntax	http://www.ietf.org/rfc/rfc3986.txt
IETF RFC 2046, Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types	http://www.ietf.org/rfc/rfc2046.txt
Naming and Addressing: URIs, URLs, and so on	http://www.w3.org/Addressing/
Resource Description Framework (RDF):	http://www.w3.org/RDF/
RDF Model and Syntax Specification	http://www.w3.org/TR/rdf-syntax-grammar/
Unicode	http://www.unicode.org/
XML 1.0 Specification	http://www.w3.org/TR/2006/REC-xml-20060816/
Namespaces in XML 1.0	http://www.w3.org/TR/2006/REC-xml-names-20060816/

1 XMP Schemas

This chapter contains the following information:

- "XMP schema definitions" on page 6: An overview of schema definitions and terminology.
- "Property value types" on page 8: Definitions and explanations of property values used by the schemas.
- "Extensibility of schemas" on page 26: Guidelines for extending XMP.

XMP schema definitions

The schemas define a set of properties. In any given XMP, a property may be *absent* or *present*:

ABSENT: The property has no value. Properties are absent until given a value for the first time.

PRESENT: The property has a defined value. A present property may have the empty string as its value; this is different from an absent property. However, writers are encouraged not to set properties with a value of the empty string.

For any given XMP, there is no requirement that all properties from a given schema must be present. For structured properties, there is no requirement that all fields be present (unless otherwise specified by a schema).

XMP metadata may include properties from one or more of the schemas. For example, a typical subset used by many Adobe applications might include the following:

Dublin Core schema: dc:title, dc:creator, dc:description, dc:subject, dc:format, dc:rights
XMP basic schema: xmp:CreateDate, xmp:CreatorTool, xmp:ModifyDate, xmp:MetadataDate
XMP rights management schema: xmpRights:WebStatement, xmpRights:Marked
XMP media management schema: xmpMM:DocumentID

Included schemas

The following schemas definitions are included in this document:

- Chapter 2, "XMP Standard Schemas"

  Dublin Core schema
  XMP Basic schema
  XMP Rights Management schema
  XMP Media Management schema
  XMP Basic Job Ticket schema
  XMP Paged-text schema
  XMP Dynamic Media schema

- Chapter 3, "Specialized Schemas"

  Adobe PDF schema
  Photoshop schema

Camera Raw schema
EXIF schemas

NOTE: This document does not provide details of the IPTC schema. For complete information on this schema, see the IPTC Web site at http://www.iptc.org/IPTC4XMP/.

Schema definition conventions

The schema definitions in this document show the XML namespace URI that identifies the schema, and a preferred schema namespace prefix, followed by a table that lists all properties defined for the schema. Each table has the following columns:

- **Property** — the name of the property, including the preferred namespace prefix.
- **Value type** — The value type of the property, with links to where each value type is described in "Property value types" on page 8. Array types are preceded by the container type: `alt`, `bag`, or `seq`; see *XMP Specification Part 1, Data and Serialization Models* for details.
- **Category** — Schema properties are *internal* or *external*:
  - Internal metadata must be maintained by an application. It can include system-level information (such as modification date) or information that an editing application has access to (such as the number of words in a document). An example is xmp:ModifyDate. Users should not be allowed to change the values of such properties. When a file is saved, an application should provide valid values for all internal properties. If an application does not set the value of an internal property, it should discard any value that may have existed previously.
  - External metadata must be set by a user, and is independent of the contents of the document. External modifications should be displayed by the editing application but are not acted upon. Unless changed by the user, external properties are preserved on output. An example is dc:creator.
- **Description** — The description of the property.

Some XMP properties have been deprecated since earlier versions of the specification. They are defined here for compatibility purposes, but should not be used in the future.

Previous versions of this specification referred to *aliased* properties. Specific XMP implementations may treat a property in one schema as equivalent to a property in another schema. However, to foster interchange, applications must always write the standard, "base" form of the property. In this version of the specification, only the base properties are listed.

An item in the xmp:Identifier array may be qualified with xmpidq:Scheme to denote the formal identification system to which that identifier conforms.

- The qualifier namespace URI is `http://ns.adobe.com/xmp/Identifier/qual/1.0/`
- The preferred qualifier namespace prefix is `xmpidq`

Property value types

The following tables list the value types used in the XMP schemas.

Basic value types

Boolean

Allowed values are `True` or `False` (the strings should be spelled exactly as shown).

Choice

A value chosen from a *vocabulary* of values, and represented by a string. Vocabularies provide a means of specifying a limited but extensible set of values for a property. The metadata schema can indicate whether the set of legal values is fixed or can be extended.

A choice can be *open* or *closed*:

- An open choice has one or more lists of preferred values, but other values can be freely used.
- A closed choice only allows values from the defined lists.

If a property value is to have a very definite meaning and all users of that property must know the exact meaning, use a closed choice vocabulary. If there are well-defined sets of values whose meanings are known, but additional values might be used without causing problems, use an open choice.

Colorant

A structure containing the characteristics of a colorant (swatch) used in a document.

- The field namespace URI is `http://ns.adobe.com/xap/1.0/g/`
- The preferred field namespace prefix is `xmpG`

Field name	Value type	Description
`xmpG:A` `xmpG:B`	Integer	A or B value when the `mode` is `LAB`. Range -128 to 127.
`xmpG:L`	Real	L value when the `mode` is `LAB`. Range 0-100.

`xmpG:black`
`xmpG:cyan`
`xmpG:magenta`
`xmpG:yellow`
	Real	Color value when the `mode` is `CMYK`. Range 0-100.

`xmpG:blue`
`xmpG:green`
`xmpG:red`
	Integer	Color value when the `mode` is `RGB`. Range 0-255.

`xmpG:mode`	closed Choice	The color space in which the color is defined. One of: `CMYK`, `RGB`, `LAB`. Library colors are represented in the color space for which they are defined.

Field name	Value type	Description
xmpG:swatchName	Text	Name of the swatch.
xmpG:type	closed Choice	The type of color, one of PROCESS or SPOT.

Date

A date-time value which is represented using a subset of ISO RFC 8601 formatting, as described in http://www.w3.org/TR/Note-datetime.html. The following formats are supported:

```
YYYY
YYYY-MM
YYYY-MM-DD
YYYY-MM-DDThh:mmTZD
YYYY-MM-DDThh:mm:ssTZD
YYYY-MM-DDThh:mm:ss.sTZD
```

YYYY = four-digit year
MM = two-digit month (01=January)
DD = two-digit day of month (01 through 31)
hh = two digits of hour (00 through 23)
mm = two digits of minute (00 through 59)
ss = two digits of second (00 through 59)
s = one or more digits representing a decimal fraction of a second
TZD = time zone designator (Z or +hh:mm or -hh:mm)

The time zone designator is optional in XMP. When not present, the time zone is unknown, and software should not assume anything about the missing time zone.

It is recommended, when working with local times, that you use a time zone designator of +hh:mm or -hh:mm instead of Z, to aid human readability. For example, if you know a file was saved at noon on October 23 a timestamp of 2004-10-23T12:00:00-06:00 is more understandable than 2004-10-23T18:00:00Z.

Dimensions

A structure containing dimensions for a drawn object.

The field namespace URI is http://ns.adobe.com/xap/1.0/sType/Dimensions#

The preferred field namespace prefix is stDim

Field name	Value type	Description
stDim:h stDim:w	Real	Height and width magnitude.
stDim:unit	open Choice	Units. For example: inch, mm, pixel, pica, point

Font

A structure containing the characteristics of a font used in a document.

- The field namespace URI is `http:ns.adobe.com/xap/1.0/sType/Font#`
- The preferred field namespace prefix is `stFnt`

Field name	Value type	Description
`stFnt:childFontFiles`	Seq String	The list of file names for the fonts that make up a composite font.
`stFnt:composite`	Boolean	When `true`, this is a composite font.
`stFnt:fontFace`	Text	The font face name.
`stFnt:fontFamily`	Text	The font family name.
`stFnt:fontFileName`	String	The font file name (not a complete path).
`stFnt:fontName`	Text	Postscript name of the font.
`stFnt:fontType`	open Choice	The font type, such as `TrueType`, `Type 1`, `Open Type`, and so on.
`stFnt:versionString`	String	The version string: - `/version` for Type1 fonts - `nameId 5` for Apple True Type and OpenType - `/CIDFontVersion` for CID fonts - The empty string for bitmap fonts CoolType allows two fonts with the same Postscript name and different technologies to be used at the same time, but not if they are from different versions. So even without this data for a given document you will have unique font data. However, the version can tell you if the font has changed metrics, glyph forms or other important information. This is useful for comparing fonts in two documents or fonts in a document to those in your system.

Integer

A signed or unsigned numeric string used as an integer number representation. The string consists of an arbitrary length decimal numeric string with an optional leading "+" or "–" sign.

Lang Alt

A *language alternative* (see *XMP Specification Part 1, Data and Serialization Models*), which is an array of type "alt Text", an alternative array of text items each of which has a language qualifier.

Locale

A closed choice that identifies a language, with values from RFC 3066.

MIMEType

A text value that identifies the file format. MIME types are defined in RFC 2046.

ProperName

A name of a person or organization, represented as a Unicode text string.

Real

A numeric value of arbitrary precision. Consists of a decimal numeric string with an optional single decimal point and an optional leading "+" or "–" sign.

It can optionally have the qualifier `vQual:binRep`, of type Text, which provides an alternate binary representation for the number when an exact value is needed. The text is interpreted as:

`std size,endian,hexadecimal_value`

- *`std`* is the standard name (`"IEEE754"`)
- *`size`* is `S` for 32-bit and `D` for 64-bit
- *`endian`* is `L` for little-endian, `B` for big-endian.

For example: `"IEEE754D,L,3A4901F387D31108"`

Text

A Unicode string.

Thumbnail

A thumbnail image for a file.

- The field namespace URI is `http://ns.adobe.com/xap/1.0/g/img/`
- The preferred field namespace prefix is `xmpGImg`

Field name	Value type	Description
`xmpGImg:format`	Closed Choice	The image encoding. Defined value: `JPEG`.

Field name	Value type	Description
xmpGImg:height xmpGImg:width	Integer	Height and width in pixels
xmpGImg:image	Text	The full thumbnail image data, converted to base 64 notation (according to section 6.8 of RFC 2045). This is the thumbnail data typically found in a digital image, such as the value of tag 513 in a JPEG stream.

URI

An Internet Uniform Resource Identifier: a compact string of characters for identifying an abstract or physical resource. See http://www.w3.org/Addressing/.

NOTE: The use of pathname properties can have privacy implications, as paths can include identifying information such as personal or company names. Applications may wish to provide appropriate user-level controls when displaying or modifying such properties.

URL

An Internet Uniform Resource Locator. See http://www.w3.org/Addressing/. An informal term (no longer used in technical specifications) associated with popular URI schemes: `http`, `ftp`, `mailto`, and so on.

XPath

XML Path Language (XPath), for addressing parts of an XML document; see http://www.w3.org/TR/xpath.

Media management value types

AgentName

The name of a program, a Text value. It is recommended that the value use this format convention:

`Organization Software_name Version (token;token;...)`

`Organization`: The name of the company or organization providing the software, no spaces.
`Software_name`: The full name of the software, spaces allowed.
`version`: The version of the software, no spaces.
`tokens`: Can be used to identify an operating system, plug-in, or more detailed version information.

For example: "Adobe Acrobat 9.0 (MacOS X 10.5)"

Part

Identifies a portion of a document. This can be a position at which the document has been changed since the most recent event history (stEvt:changed). For a resource within an xmpMM:Ingredients list, the ResourceRef uses this type to identify both the portion of the containing document that refers to the resource, and the portion of the referenced resource that is referenced.

Part names are a hierarchy of arbitrary depth, specified using path syntax where levels in the hierarchy are indicated by the slash "/" character. The slash may not be used for any other purpose in these strings. Paths (including partial paths) must always start from "/" (meaning *all* or *root*).

If a partial path is specified, it is assumed to encompass all further descendents of the last level specified. For example, `/metadata` includes all descendants of metadata, whereas `/metadata/crs` includes all camera raw settings, but excludes metadata that is not descendent from camera raw settings. Additional levels of sub-parts or alternatives for existing levels may be defined; for example `/content/audio/channels/left` or `/content/audio/FFTaudio/high`. When such subparts are defined, each subpart name must be unique and signify a component that is disjoint from any of its siblings.

A new part component can be added at any level; it is assumed to be something new and disjoint from other children of its parent path.

Part component names in the hierarchy normally consist of letters, digits, and a limited number of punctuation characters (period, hyphen, underscore, and colon). Formally, each part component must follow the syntax of XML "Name" (with or without a namespace prefix), as specified in the W3C-XML-1.1 and W3C-XML-NAMES specifications; the colon has special meaning indicating a namespace prefix. All other punctuation characters (below U+007f) are reserved for future use; software should interpret path strings with a "/"-delimited component containing reserved characters by ignoring that component and its descendents (that is, indicating that the parent component has changed, with any restrictions to subparts unknown). As with XML names, although full Unicode names are supported (and the names given are often meaningful English words) the elements of a part name hierarchy are string tokens not intended to be translated.

The following are explicitly defined:

Part specification	Part that changed or is referenced
/	Any (specific part unknown) or all (all parts of the content and metadata).
`/metadata`	Portions of the metadata.
`/content`	Any or all of the content (non-metadata).
`/content/audio`	Any or all sound.
`/content/visual`	Some image data (video or still).
`/content/visual/video`	Video or animation.
`/content/visual/raster`	Static raster image.
`/content/visual/vector`	Static vector image.
`/content/visual/form/data`	Form field data.
`/content/visual/form/template`	Form template.

Part specification	Part that changed or is referenced
`/content/visual/annots`	Applied annotations (comments).

`[/]time:##`
`[/]time:##d##`
`[/]time:##r##`
	A time, duration, or time range specifier. May be standalone (meaning all parts starting at the time or within the range specified) or may be added to any of the listed specifications.
	
	`##`: The start time, a frame count.
	`##d##`: Duration (start time and duration time)
	`##r##`: Range (start time and end time)
	
	Each `##` value is a FrameCount specifier, which can include an optional frame rate. The default frame rate is 1fps. The default duration is `"maximum"`, the entire length of the asset.
	
	➤ In a `fromPart` or `toPart` value, the leading `/` is optional. For an `stEvt:changed` part descriptor in a history record, the leading `/` is required.
	
	➤ For a `fromPart` value, the start time is an offset from the start of the current ingredient's file. For a `toPart` value, the start time is measured from the start of the destination file. If time values are not specifically given, the default start time is 0, meaning the beginning of the relevant file.

RenditionClass

The type of a rendition, from a controlled vocabulary of standard names (an open Choice). A series of colon-separated tokens and parameters, the first of which names the basic concept of the rendition. Additional tokens are optional and provide specific characteristics of the rendition. Defined values are:

`default`	The master document; no additional tokens allowed.
`draft`	A review rendition.
`low-res`	A low resolution, full size stand-in.
`proof`	A review proof.
`screen`	Screen resolution or Web rendition.
`thumbnail`	A simplified or reduced preview of a version. Additional tokens can provide characteristics. The recommended order is: `thumbnail`:*format*:*size*:*colorspace*. For example: `thumbnail:jpeg`, `thumbnail:16x16`, `thumbnail:gif:8x8:bw`.

ResourceEvent

A high-level event that occurred in the processing of this document.

➤ The field namespace URI is `http://ns.adobe.com/xap/1.0/sType/ResourceEvent#`

➤ The preferred field namespace prefix is `stEvt`

Field name	Value type	Description
stEvt:action	open Choice	The action that occurred. Defined values are: converted, copied, created, cropped, edited, filtered, formatted, version_updated, printed, published, managed, produced, resized, saved New values should be verbs in the past tense.
stEvt:changed	list of Part	Optional. A semicolon-delimited list of the parts of the document that were changed since the previous event history. The part names are a hierarchy of any depth. If not present, presumed to be undefined. When tracking changes, a conservative approach is that if the scope of the changed components is unknown, it must be assumed that anything might have changed.
stEvt:instanceID	URI	The instance ID of the modified resource.
stEvt:parameters	Text	Additional description of the action.
stEvt:softwareAgent	AgentName	The software agent that performed the action.
stEvt:when	Date	Optional timestamp of when the action occurred. For events that create or write a file, this should be the (estimated) modification time of the file. If the file system's actual modification date is later than this value, some XMP software may assume the file was modified by a non-XMP-aware application.

ResourceRef

A multiple part reference to a resource. Used to indicate prior versions, originals of renditions, originals for derived documents, and so on. The fields present in any specific reference depend on usage and on whether the referenced resource is managed. Except for instanceID, the fields are all properties from the referenced resource's xmpMM schema.

- The field namespace URI is http://ns.adobe.com/xap/1.0/sType/ResourceRef#
- The preferred field namespace prefix is stRef

Field name	Value type	Description
stRef:alternatePaths	Seq of URI	The referenced resource's fallback file paths or URLs. The sequence order is the recommended order in attempting to locate the resource.
stRef:documentID	URI	The referenced resource's xmpMM:DocumentID.
stRef:filePath	URI	The referenced resource's file path or URL.

Field name	Value type	Description
stRef:fromPart	Part	For a resource within an xmpMM:Ingredients list, the part of this resource that is incorporated in the containing document.
stRef:instanceID	URI	The referenced resource's xmpMM:InstanceID.
stRef:lastModifyDate	Date	The value of stEvt:when for the last time the file was written.
stRef:manager	AgentName	The referenced resource's xmpMM:Manager.
stRef:managerVariant	Text	The referenced resource's xmpMM: ManagerVariant.
stRef:manageTo	URI	The referenced resource's xmpMM:ManageTo.
stRef:manageUI	URI	The referenced resource's xmpMM:ManageUI.
stRef:maskMarkers	closed Choice	For a resource within an xmpMM:Ingredients list, whether markers in this resource should be ignored (masked) or processed normally. One of: All: Ignore markers in this ingredient and all its children. None: Process markers in this ingredient and all its children.
stRef:partMapping	Text	The name or URI of a mapping function used to map the fromPart to the toPart. The default for time mappings is "linear".
stRef:renditionClass	RenditionClass	The referenced resource's xmpMM: RenditionClass.
stRef:renditionParams	Text	The referenced resource's xmpMM: RenditionParams.
stRef:toPart	Part	For a resource within an xmpMM:Ingredients list, the part of the containing document into which this resource is incorporated.
stRef:versionID	Text	The referenced resource's xmpMM:VersionID.

Version

Describes one version of a document.

- The field namespace URI is http://ns.adobe.com/xap/1.0/sType/Version#
- The preferred field namespace prefix is stVer

Field name	Value type	Description
stVer:comments	Text	Comments concerning what was changed.
stVer:event	ResourceEvent	High level, formal description of what operation the user performed.

Field name	Value type	Description
stVer:modifier	ProperName	The person who modified this version.
stVer:modifyDate	Date	The date on which this version was checked in.
stVer:version	Text	The new version number.

Basic job/workflow value types

The following value type is used for the Basic Job/Workflow schema.

Job

Describes a job for a job-management system.

- The field namespace URI is `http://ns.adobe.com/xap/1.0/sType/Job#`
- The field namespace prefix is `stJob`

Field name	Value type	Description
stJob:id	Text	Unique ID for the job. This field is a reference into some external job management system.
stJob:name	Text	Informal name of job. This name is for user display and informal systems.
stJob:url	URL	A file URL referencing an external job management file.

Video media value types

The following value types are used for the XMP Dynamic Media schema.

- The field namespace URI is `http://ns.adobe.com/xmp/1.0/DynamicMedia/`
- The preferred field namespace prefix is `xmpDM`

beatSpliceStretch

A set of parameters used when stretching audio using the `Beat Splice` stretch mode.

Field name	Value type	Description
xmpDM:riseInDecibel	Real	The amount sound must increase in amplitude to detect a beat.
xmpDM:riseInTimeDuration	Time	The duration of the sampling window used to measure the audio increase for locating beats.
xmpDM:useFileBeatsMarker	Boolean	When `true`, the file beat markers are used for stretching. Otherwise the rise and duration fields are used to automatically locate the beats.

CuePointParam

A key-value pair describing a parameter of a cue-point Marker.

Field name	Value type	Description
xmpDM:key	Text	The key.
xmpDM:value	Text	The value.

FrameCount

A number of frames at a given frame rate, which specifies an audio or video time value for a Marker (as the value of xmpDM:duration or xmpDM:startTime). Can also be used in the time portion of a document Part.

The frame-count value can include the frame rate as shown. For Markers within a Track, the frame rate can be specified separately in the xmpDM:frameRate of the Track.

The string value is in one of these formats:

"##"	For a Marker that is not in a Track, a simple integer value is interpreted as a number of seconds, at the default frame rate of 1 fps. For Markers within a Track, an integer value is interpreted as ticks/frames in the timescale specified by the track's xmpDM:frameRate. When the count is zero, no frame rate should be specified.
"##f###" "##f###s###"	A number of frames specified together with a FrameRate, which can contain an optional rate basis. The rate basis defaults to 1. These examples show how a FrameCount value of 15 is expressed for common video and audio frame rates: ➤ Film at 24 fps (frame rate = 24, rate basis = 1): "15f24" ➤ Speech-to-text in milliseconds (frame rate = 1000, rate basis = 1): "15f1000" ➤ NTSC at 29.92 fps (frame rate = 30000, rate basis = 1001): "15f30000s1001" ➤ DVATicks (frame rate = 254016000000, rate basis = 1): "15f254016000000"
"maximum"	Allowed for a duration value; indicates that the time span is unlimited, or is determined automatically up to the full duration of the source.

FrameRate

A frame-rate value can be part of the FrameCount specification of a Marker. For Markers within a Track, however, the frame count can be a simple integer, and the associated frame rate is specified separately, in the xmpDM:frameRate of the Track.

A frame rate is expressed as a number of frames divided by a number of seconds (f/s). The number of seconds is called the *rate basis*; it defaults to 1, for the common frames-per-second (fps) expression. If no

frame rate is specified in either the frame count itself or in the associated track, the frame count is also the number of seconds, at the default rate of 1 fps.

The string value is in one of these formats:

`"f###"`	The frame rate in frames-per-second (fps). The rate basis is assumed to be 1. For example, a frame rate of 24fps is specified as "f24".
`"f###s###"`	Specifies a frame rate with a rate basis. The second number is the rate basis, a number of seconds. For example, the NTSC 29.97 frame rate is specified as "f30000s1001".

Marker

A marker type used to describe an important location in an audio or video sequence. It is a value of the `xmpDM:markers` array in a Track.

Field name	Value type	Description
`xmpDM:comment`	Text	Optional. A descriptive comment.
`xmpDM:cuePointParams`	Seq of CuePointParam	Optional. An ordered sequence of processing parameters for an FLVCuePoint-type marker.
`xmpDM:cuePointType`	Text	Optional. The cue-point type for an FLVCuePoint-type marker, one of `Navigation` or `Event`.
`xmpDM:duration`	FrameCount	Optional. The duration of the marker. Default is 0. This is a number of ticks/frames in the timescale specified by an optionally included frame rate. If the parent Track specifies xmpDM:frameRate, that becomes the default frame rate for all member markers. If no frame rate is specified in either the marker or the track, this value is a number of seconds (frames at the default frame rate of 1fps).
`xmpDM:location`	URI	Optional. The URL of the location to jump to, for a WebLink-type marker. For example, `http://www.mysite.com`.
`xmpDM:name`	Text	The name of the marker. For timed text, the phrase, word, or syllable.
`xmpDM:probability`	Real	Optional. For auto-detected speech, the probability that the word is accurate.
`xmpDM:speaker`	Text	Optional. The name or other identifier of the speaker or performer, for a Speech-type marker. The speaker need only be identified if it changes from the previous phrase.

Field name	Value type	Description
xmpDM:startTime	FrameCount	The timeline position of the marker. Default is 0, the beginning of the file that contains the track.
xmpDM:target	Text	Optional. A frame target, for a WebLink-type marker.

xmpDM:type	Open Choice of Text, comma-delimited list	Optional. A comma-delimited list of marker types. The type indicates how a marker or set of markers is intended to be used, and what other information is associated with it. Predefined value types include:
		`Chapter`
		`Cue`
		`Index`
		`Speech`
		`Track`
		This type overrides any type specified in the containing Track.

Media

A reference to a media asset. This is typically a local file, but can be anything that can be specified with a URL. Contains information about usage in the parent media (typically a sequence), and the associated media rights.

Field name	Value type	Description
xmpDM:duration	Time	The duration of the asset in the timeline.
xmpDM:managed	Boolean	When `true`, this is a rights-managed resource.
xmpDM:path	URI	The location of the asset.
xmpDM:startTime	Time	The timeline position of the start of the asset, an offset from the beginning of the file that contains the track.
xmpDM:track	Text	An identifier for the track that contained this asset. Could be a track name or a number.
xmpDM:webStatement	URI	The location of a web page describing the owner and/or rights statement for this resource.

ProjectLink

The type of a video file and path of the project that created it.

Field name	Value type	Description
xmpDM:path	URI	Full path to the project that created this file.
xmpDM:type	Closed Choice of Text	The file type. One of: movie still audio custom

resampleStretch

A set of parameters used when stretching audio using the Resample stretch mode.

Field name	Value type	Description
xmpDM:quality	Closed Choice of Text	One of: High Medium Low

Time

A representation of a time value in seconds. This is similar to After Effect's TDB, or QuickTime's representation of time. They each have a value, and the scale of the value. For example, if the scale is the rational 1/25 (PAL 25fps), and the value is 50, the time is 2 seconds.

Field name	Value type	Description
xmpDM:scale	Rational	The scale for the time value. ➤ For NTSC, use 1001/30000, or the less accurate 100/2997. ➤ For PAL, use 1/25.
xmpDM:value	Integer	The time value in the specified scale.

Timecode

A time value in video.

Field name	Value type	Description
xmpDM:timeFormat	Closed Choice of Text	The format used in the timeValue. One of: 24Timecode 25Timecode 2997DropTimecode (*semicolon delimiter*) 2997NonDropTimecode 30Timecode 50Timecode 5994DropTimecode 5994NonDropTimecode 60Timecode 23976Timecode
xmpDM:timeValue	Text	A time value in the specified format. Time values use a colon delimiter in all formats except 2997drop, which uses a semicolon. The four fields indicate hours, minutes, seconds, and frames: *hh:mm:ss:ff* The actual duration in seconds depends on the format.
xmpDM:value	Integer	The time value in the specified scale.

timeScaleStretch

A set of parameters used when stretching audio using the Time-Scale stretch mode.

Field name	Value type	Description
xmpDM:frameOverlappingPercentage	Real	The percentage of overlap between frames.
xmpDM:frameSize	Real	The splices per beat.
xmpDM:quality	Closed Choice of Text	One of: High Medium Low

Track

A named set of Markers, that can specify different default time-frame rates from those of the contained markers.

xmpDM:frameRate	FrameRate	The default frame rate for the markers in the track.
xmpDM:markers	Seq of Marker	An ordered list of markers.

xmpDM:trackName	Text	The name of the track. (For example: Lyrics, Speech, Voiceover, Audition Conditions, and so on.)
xmpDM:trackType	Open Choice of Text, comma-delimited list	The default marker types for all markers in the track. See Marker field xmpDM:type.

EXIF schema value types

These types are used only within the EXIF schema.

CFAPattern

A structure describing the CFA pattern.

- The field namespace URI is http://ns.adobe.com/exif/1.0/
- The preferred field namespace prefix is exif

Field name	Value type	Description
exif:Columns	Integer	Number of columns, *n*.
exif:Rows	Integer	Number of rows, *m*.
exif:Values	seq Integer	CFA values, sequence should be, in order: value [0,0] … value [*n* −1, 0] value [0, *m* −1] … value [*n* −1, *m* −1]

DeviceSettings

A structure describing the device settings.

- The field namespace URI is http://ns.adobe.com/exif/1.0/
- The preferred field namespace prefix is exif

Field name	Value type	Description
exif:Columns	Integer	Display columns.
exif:Rows	Integer	Display rows.
exif:Settings	seq Text	Camera settings, in order.

Flash

A structure describing the flash state.

- The field namespace URI is `http://ns.adobe.com/exif/1.0/`
- The preferred field namespace prefix is `exif`

Field name	Value type	Description
`exif:Fired`	Boolean	`True` if flash fired.
`exif:Function`	Boolean	`True` if flash function is not present.
`exif:Mode`	Closed Choice	The flash mode. One of: 0 = unknown 1 = compulsory flash firing 2 = compulsory flash suppression 3 = auto mode
`exif:RedEyeMode`	Boolean	`True` if red-eye reduction is supported.
`exif:Return`	Closed Choice	Whether strobe return is supported and if supported, detected. One of: 0 = no strobe return detection 2 = strobe return light not detected 3 = strobe return light detected

GPSCoordinate

A Text value in the form "*DDD*,*MM*,*SSk*" or "*DDD*,*MM*.*mmk*", where:

DDD is a number of degrees
MM is a number of minutes
SS is a number of seconds
mm is a fraction of minutes
k is a single character `N`, `S`, `E`, or `W` indicating a direction (north, south, east, west)

Leading zeros are not necessary for the for *DDD*, *MM*, and *SS* values. The *DDD*,*MM*.*mmk* form should be used when any of the native EXIF component rational values has a denominator other than 1. There can be any number of fractional digits.

OECF/SFR

A structure describing the OECF/SFR.

- The field namespace URI is `http://ns.adobe.com/exif/1.0/`
- The preferred field namespace prefix is `exif`

Field name	Value type	Description
exif:Columns	Integer	Number of columns, *n*.
exif:Names	seq Text	Column item names, *n* entries.
exif:Rows	Integer	Number of rows, *m*.
exif:Values	seq Rational	OECF/SFR values, sequence should be, in order: value [0,0] … value [*n* −1, 0] value [0, *m* −1] … value [*n* −1, *m* −1]

Rational

To represent EXIF rational values in XMP, they must be converted to text. The recommended approach is to use a value of type Text of the form *numerator*/*denominator*. For example, the value 2/3 becomes the text value "2/3" when converted to XMP.

Extensibility of schemas

This section discusses how to create new schemas and extend existing ones.

Creating custom schemas

The schemas defined in this document are core schemas that are believed to be applicable to a wide variety of needs. If possible, it is always desirable to use properties from existing schemas. However, XMP was designed to be easily extensible by the addition of custom schemas. If your metadata needs are not already covered by the core schemas, you can define and use your own schemas.

If you are considering creating a new namespace, observe the following:

- Avoid including properties that have the same semantics as properties in existing namespaces.
- If your properties might be useful to others, try to collaborate in creating a common namespace, to avoid having a multitude of incompatible ones.

To define a new schema, you should write a human-readable schema specification document. The specification document should be made available to any developers who need to write code that understands your metadata. (Future versions of XMP might include support for machine-readable schema specifications, but such support will always be in addition to the requirement for human-readable schema specification documents.)

Your specification document should include:

- A unique name for your schema in the form of a URI and a preferred prefix.
- A table containing the name of each property, the value type, and the description of the property. If you define properties that have structured value types, you may wish to use additional URI names to identify the components of a structured property value.

You can then add more properties as needed, following the RDF and XMP syntax requirements described in this document to create compatible RDF metadata.

Extending schemas

Keep in mind the following points when extending a schema:

- New properties can be added to existing namespaces without causing problems for applications.
- The definitions of properties in existing namespaces should always remain the same; otherwise, applications can produce incorrect behavior. If it is necessary to change the meaning of a property, a new property should be created, and the old one declared as deprecated.
- It is possible to create a "new version" of a schema namespace; however, there is no logical connection between the old version and the new version. The same local name in two different XML namespaces refers to two different properties.

2 XMP Standard Schemas

This chapter contains the following information schema definitions for standard schemas.

The following schemas definitions are included here:

Dublin Core schema

The Dublin Core schema provides a set of commonly used properties.

- The schema namespace URI is `http://purl.org/dc/elements/1.1/`
- The preferred schema namespace prefix is `dc`

Property	Value type	Category	Description
`dc:contributor`	bag ProperName	External	Contributors to the resource (other than the authors).
`dc:coverage`	Text	External	The extent or scope of the resource.
`dc:creator`	seq ProperName	External	The authors of the resource (listed in order of precedence, if significant).
`dc:date`	seq Date	External	Date(s) that something interesting happened to the resource.
`dc:description`	Lang Alt	External	A textual description of the content of the resource. Multiple values may be present for different languages.
`dc:format`	MIMEType	Internal	The file format used when saving the resource. Tools and applications should set this property to the save format of the data. It may include appropriate qualifiers.
`dc:identifier`	Text	External	Unique identifier of the resource.
`dc:language`	bag Locale	Internal	An unordered array specifying the languages used in the resource.
`dc:publisher`	bag ProperName	External	Publishers.
`dc:relation`	bag Text		Relationships to other documents.

Property	Value type	Category	Description
dc:rights	Lang Alt	External	Informal rights statement, selected by language.
dc:source	Text	External	Unique identifier of the work from which this resource was derived.
dc:subject	bag Text	External	An unordered array of descriptive phrases or keywords that specify the topic of the content of the resource.
dc:title	Lang Alt	External	The title of the document, or the name given to the resource. Typically, it will be a name by which the resource is formally known.
dc:type	bag open Choice	External	A document type; for example, novel, poem, or working paper.

XMP Basic schema

The XMP Basic schema contains properties that provide basic descriptive information.

- The schema namespace URI is `http://ns.adobe.com/xap/1.0/`
- The preferred schema namespace prefix is `xmp`

Property	Value type	Category	Description
xmp:Advisory	bag XPath	External	An unordered array specifying properties that were edited outside the authoring application. Each item should contain a single namespace and XPath separated by one ASCII space (U+0020).
xmp:BaseURL	URL	Internal	The base URL for relative URLs in the document content. If this document contains Internet links, and those links are relative, they are relative to this base URL. This property provides a standard way for embedded relative URLs to be interpreted by tools. Web authoring tools should set the value based on their notion of where URLs will be interpreted.
xmp:CreateDate	Date	Internal	The date and time the resource was originally created.
xmp:CreatorTool	AgentName	Internal	The name of the first known tool used to create the resource. If history is present in the metadata, this value should be equivalent to that of xmpMM:History's `softwareAgent` property.
xmp:Identifier	bag Text	External	An unordered array of text strings that unambiguously identify the resource within a given context. An array item may be qualified with xmpidq:Scheme to denote the formal identification system to which that identifier conforms. The dc:identifier property is not used because it lacks a defined scheme qualifier and has been defined in the XMP Specification as a simple (single-valued) property.

Property	Value type	Category	Description
xmp:Label	Text	External	A word or short phrase that identifies a document as a member of a user-defined collection. Used to organize documents in a file browser.
xmp:MetadataDate	Date	Internal	The date and time that any metadata for this resource was last changed. It should be the same as or more recent than xmp:ModifyDate.
xmp:ModifyDate	Date	Internal	The date and time the resource was last modified. The value of this property is not necessarily the same as the file's system modification date because it is set before the file is saved.
xmp:Nickname	Text	External	A short informal name for the resource.
xmp:Rating	Closed Choice of Integer	External	A number that indicates a document's status relative to other documents, used to organize documents in a file browser. Values are user-defined within an application-defined range.
xmp:Thumbnails	alt Thumbnail	Internal	An alternative array of thumbnail images for a file, which can differ in characteristics such as size or image encoding.

An item in the xmp:Identifier array may be qualified with xmpidq:Scheme to denote the formal identification system to which that identifier conforms.

- The qualifier namespace URI is `http://ns.adobe.com/xmp/Identifier/qual/1.0/`
- The preferred qualifier namespace prefix is `xmpidq`

Qualifier	Value type	Category	Description
xmpidq:Scheme	Text	External	The name of the formal identification system used in the value of the associated xmp:Identifier item.

XMP Rights Management schema

This schema includes properties related to rights management. These properties specify information regarding the legal restrictions associated with a resource.

NOTE: XMP is not a rights-enforcement mechanism.

- The schema namespace URI is `http://ns.adobe.com/xap/1.0/rights/`
- The preferred schema namespace prefix is `xmpRights`

Property	Value type	Category	Description
xmpRights:Certificate	URL	External	Online rights management certificate.
xmpRights:Marked	Boolean	External	Indicates that this is a rights-managed resource.
xmpRights:Owner	bag ProperName	External	An unordered array specifying the legal owner(s) of a resource.
xmpRights:UsageTerms	Lang Alt	External	Text instructions on how a resource can be legally used.
xmpRights:WebStatement	URL	External	The location of a web page describing the owner and/or rights statement for this resource.

XMP Media Management schema

This schema is primarily for use by digital asset management (DAM) systems.

The following properties are "owned" by the DAM system and should be set by applications under their direction; they should not be used by unmanaged files: xmpMM: ManagedFrom, xmpMM:Manager, xmpMM:ManageTo, xmpMM:ManageUI, xmpMM: ManagerVariant.

The following properties are owned by the DAM system for managed files, but can also be used by applications for unmanaged files: xmpMM:DerivedFrom, xmpMM:DocumentID, xmpMM: RenditionClass, xmpMM: RenditionParams, xmpMM:VersionID, xmpMM:Versions.

The xmpMM:History property is always owned by the application.

- The schema namespace URI is `http://ns.adobe.com/xap/1.0/mm/`
- The preferred schema namespace prefix is `xmpMM`

Property	Value type	Category	Description
xmpMM:DerivedFrom	ResourceRef	Internal	A reference to the original document from which this one is derived. It is a minimal reference; missing components can be assumed to be unchanged. For example, a new version might only need to specify the instance ID and version number of the previous version, or a rendition might only need to specify the instance ID and rendition class of the original.
xmpMM:DocumentID	URI	Internal	The common identifier for all versions and renditions of a document. See Document and instance IDs below.
xmpMM:History	seq ResourceEvent	Internal	An ordered array of high-level user actions that resulted in this resource. It is intended to give human readers a description of the steps taken to make the changes from the previous version to this one. The list should be at an abstract level; it is not intended to be an exhaustive keystroke or other detailed history. The description should be sufficient for metadata management, as well as workflow enhancement.
xmpMM:Ingredients	Bag ResourceRef	Internal	An unordered array of references to resources that were incorporated, by inclusion or reference, into this document.
xmpMM:InstanceID	URI	Internal	An identifier for a specific incarnation of a document, updated each time a file is saved. It should be based on a UUID; see Document and instance IDs below.

Property	Value type	Category	Description
xmpMM: ManagedFrom	ResourceRef	Internal	A reference to the document as it was prior to becoming managed. It is set when a managed document is introduced to an asset management system that does not currently own it. It may or may not include references to different management systems.
xmpMM:Manager	AgentName	Internal	The name of the asset management system that manages this resource. Along with xmpMM: ManagerVariant, it tells applications which asset management system to contact concerning this document.
xmpMM:ManageTo	URI	Internal	A URI identifying the managed resource to the asset management system; the presence of this property is the formal indication that this resource is managed. The form and content of this URI is private to the asset management system.
xmpMM:ManageUI	URI	Internal	A URI that can be used to access information about the managed resource through a web browser. It might require a custom browser plug-in.
xmpMM: ManagerVariant	Text	Internal	Specifies a particular variant of the asset management system. The format of this property is private to the specific asset management system.
xmpMM: OriginalDocumentID	URI	Internal	The common identifier for all versions and renditions of a document; see Document and instance IDs below.
xmpMM:Pantry	Bag struct	Internal	Each array item is a struct with a potentially unique set of fields, containing the full XMP from a component. Each field is a top level property from the XMP of a contained document component, with all substructure preserved. Each pantry entry must contain an xmpMM:InstanceID. Only one copy of the pantry entry for any given instance ID should be retained in the pantry. Nested pantry items are removed from the individual pantry item and promoted to the top level of the pantry.

Property	Value type	Category	Description
xmpMM: RenditionClass	RenditionClass	Internal	The rendition class name for this resource. This property should be absent or set to `default` for a document version that is not a derived rendition.
xmpMM: RenditionParams	Text	Internal	Can be used to provide additional rendition parameters that are too complex or verbose to encode in xmpMM: RenditionClass.
xmpMM:VersionID	Text	Internal	The document version identifier for this resource. Each version of a document gets a new identifier, usually simply by incrementing integers 1, 2, 3 . . . and so on. Media management systems can have other conventions or support branching which requires a more complex scheme.
xmpMM:Versions	seq Version	Internal	The version history associated with this resource. Entry `[1]` is the oldest known version for this document, entry `[last()]` is the most recent version. Typically, a media management system would fill in the version information in the metadata on check-in. It is not guaranteed that a complete history of versions from the first to this one will be present in the `xmpMM:Versions` property. Interior version information can be compressed or eliminated and the version history can be truncated at some point.
xmpMM:LastURL (*deprecated*)	URL	Internal	Deprecated for privacy protection.
xmpMM:RenditionOf (*deprecated*)	ResourceRef	Internal	Deprecated in favor of xmpMM:DerivedFrom. A reference to the document of which this is a rendition.
xmpMM:SaveID (*deprecated*)	Integer	Internal	Deprecated. Previously used only to support the `xmpMM:LastURL` property.

Document and instance IDs

There can often be ambiguity when referring to computer files. The contents of a file can change over time. Depending on the situation, it might be desirable to refer to either:

- a *specific* state of the file as it exists at a point in time, or

- the file in general, as a *persistent* container whose content can change.

Some characteristics of a file (such as the application that created it) would normally be expected to be persistent over its life. Other characteristics (such as word count) would be expected to change as the content of the file changes. Some characteristics (such as copyright information or authors' names) might or might not change.

In the same way, XMP properties that represent such characteristics of a file are inherently ambiguous as to whether they refer to the current content of a file or to the file in general. XMP itself provides no mechanisms for distinguishing these. Schemas are encouraged, but not required, to define properties in a way that makes this clear.

This document uses the term *resource* to refer to the "thing the metadata is about" in a general sense. Depending on the context, properties may refer to either specific or persistent aspects of the described file. In order to refer unambiguously to a specific state of the file, we use the term *instance*.

> **NOTE:** This terminology should be distinguished from HTTP terminology, where *resource* is most often used in the sense of "container", while *entity* or *entity-part* is always used to mean "the current content of all or part of a resource at some point in time."

An ID should be guaranteed to be globally unique (in practical terms, this means that the probability of a collision is so remote as to be effectively impossible). Typically 128- or 144-bit numbers are used, encoded as hexadecimal strings.

XMP does not require any specific scheme for generating the unique number. There are various common schemes available for that purpose, such as:

- Using physical information such as a local Ethernet address and a high resolution clock. When creating a unique ID, applications must consider trade-offs between privacy and the desire to create an audit trail. Adobe applications favor privacy and do not include Ethernet addresses.
- Using a variety of locally unique and random data, then computing an MD5 hash value. This avoids privacy concerns about the use of Ethernet addresses. It also allows for regeneration of the ID in some cases; for example if the MD5 hash is computed using the image contents for a resource that is a digital photograph.

XMP Basic Job Ticket schema

This schema describes very simple workflow or job information.

- The schema namespace URI is `http://ns.adobe.com/xap/1.0/bj/`
- The preferred schema namespace prefix is `xmpBJ`

Property	Value type	Category	Description
`xmpBJ:JobRef`	bag Job	External	References an external job management file for a job process in which the document is being used. Use of job names is under user control. Typical use would be to identify all documents that are part of a particular job or contract. There are multiple values because there can be more than one job using a particular document at any time, and it can also be useful to keep historical information about what jobs a document was part of previously.

XMP Paged-text schema

This schema is used for text appearing on a page in a document.

- The schema namespace URI is `http://ns.adobe.com/xap/1.0/t/pg/`
- The preferred schema namespace prefix is `xmpTPg`

Property	Value type	Category	Description
xmpTPg:MaxPageSize	Dimensions	Internal	The size of the largest page in the document (including any in contained documents).
xmpTPg:NPages	Integer	Internal	The number of pages in the document (including any in contained documents).
xmpTPg:Fonts	Bag Font	Internal	An unordered array of fonts that are used in the document (including any in contained documents).
xmpTPg:Colorants	Seq Colorant	Internal	An ordered array of colorants (swatches) that are used in the document (including any in contained documents).
xmpTPg:PlateNames	Seq Text	Internal	An ordered array of plate names that are needed to print the document (including any in contained documents).

XMP Dynamic Media schema

This schema specifies properties used by the Adobe dynamic media group.

- The schema namespace URI is `http://ns.adobe.com/xmp/1.0/DynamicMedia/`
- The preferred schema namespace prefix is `xmpDM`

Property	Value type	Category	Description
xmpDM:absPeakAudioFilePath	URI	Internal	The absolute path to the file's peak audio file. If empty, no peak file exists.
xmpDM:album	Text	External	The name of the album.
xmpDM:altTapeName	Text	External	An alternative tape name, set via the project window or timecode dialog in Premiere. If an alternative name has been set and has not been reverted, that name is displayed.
xmpDM:altTimecode	Timecode	External	A timecode set by the user. When specified, it is used instead of the `startTimecode`.
xmpDM:artist	Text	External	The name of the artist or artists.
xmpDM:audioModDate	Date	Internal	The date and time when the audio was last modified.
xmpDM:audioSampleRate	Integer	Internal	The audio sample rate. Can be any value, but commonly `32000`, `41100`, or `48000`.

xmpDM:audioSampleType	closed Choice of Text	Internal	The audio sample type. One of:
			`8Int`
			`16Int`
			`32Int`
			`32Float`

xmpDM:audioChannelType	closed Choice of Text	Internal	The audio channel type. One of:
			`Mono`
			`Stereo`
			`5.1`
			`7.1`

xmpDM:audioCompressor	Text	Internal	The audio compression used. For example, `MP3`.
xmpDM:beatSpliceParams	beatSpliceStretch	Internal	Additional parameters for `Beat Splice` stretch mode.
xmpDM:composer	Text	External	The composer's name.

Property	Value type	Category	Description
xmpDM:contributedMedia	bag Media	Internal	An unordered list of all media used to create this media.
xmpDM:copyright	Text	External	The copyright information.
xmpDM:duration	Time	Internal	The duration of the media file.
xmpDM:engineer	Text	External	The engineer's name.
xmpDM:fileDataRate	Rational	Internal	The file data rate in megabytes per second. For example: "36/10" = 3.6 MB/sec
xmpDM:genre	Text	External	The name of the genre.
xmpDM:instrument	Text	External	The musical instrument.
xmpDM:introTime	Time	Internal	The duration of lead time for queuing music.
xmpDM:key	closed Choice of Text	Internal	The audio's musical key. One of: C C# D D# E F F# G G# A A# B
xmpDM:logComment	Text	External	User's log comments.
xmpDM:loop	Boolean	Internal	When true, the clip can be looped seamlessly.
xmpDM:numberOfBeats	Real	Internal	The number of beats.
xmpDM:markers	seq Marker	Internal	An ordered list of markers. See also xmpDM:Tracks.
xmpDM:metadataModDate	Date	Internal	The date and time when the metadata was last modified.
xmpDM:outCue	Time	Internal	The time at which to fade out.
xmpDM:projectRef	ProjectLink	Internal	A reference to the project that created this file.

Property	Value type	Category	Description

xmpDM:pullDown	closed Choice of Text	Internal	The sampling phase of film to be converted to video (pull-down). One of:
			`WSSWW`
			`SSWWW`
			`SWWWS`
			`WWWSS`
			`WWSSW`
			`WSSWW_24p`
			`SSWWW_24p`
			`SWWWS_24p`
			`WWWSS_24p`
			`WWSSW_24p`

xmpDM: relativePeakAudioFilePath	URI	Internal	The relative path to the file's peak audio file. If empty, no peak file exists.
xmpDM:relativeTimestamp	Time	Internal	The start time of the media inside the audio project.
xmpDM:releaseDate	Date	External	The date the title was released.
xmpDM:resampleParams	resampleStretch	Internal	Additional parameters for `Resample` stretch mode.

xmpDM:scaleType	closed Choice of Text	Internal	The musical scale used in the music. One of:
			`Major`
			`Minor`
			`Both`
			`Neither`
			`Neither` is most often used for instruments with no associated scale, such as drums.

xmpDM:scene	Text	External	The name of the scene.
xmpDM:shotDate	Date	External	The date and time when the video was shot.
xmpDM:shotLocation	Text	External	The name of the location where the video was shot. For example: `"Oktoberfest, Munich Germany"` For more accurate positioning, use the EXIF GPS values.
xmpDM:shotName	Text	External	The name of the shot or take.

Property	Value type	Category	Description
xmpDM:speakerPlacement	Text	External	A description of the speaker angles from center front in degrees. For example: "Left = -30, Right = 30, Center = 0, LFE = 45, Left Surround = -110, Right Surround = 110"
xmpDM:startTimecode	Timecode	Internal	The timecode of the first frame of video in the file, as obtained from the device control.
xmpDM:stretchMode	closed Choice of Text	Internal	The audio stretch mode. One of: Fixed length Time-Scale Resample Beat Splice Hybrid
xmpDM:tapeName	Text	External	The name of the tape from which the clip was captured, as set during the capture process.
xmpDM:tempo	Real	Internal	The audio's tempo.
xmpDM:timeScaleParams	timeScaleStretch	Internal	Additional parameters for Time-Scale stretch mode.
xmpDM:timeSignature	closed Choice of Text	Internal	The time signature of the music. One of: 2/4 3/4 4/4 5/4 7/4 6/8 9/8 12/8 other
xmpDM:trackNumber	Integer	External	A numeric value indicating the order of the audio file within its original recording.
xmpDM:Tracks	Bag Track	Internal	An unordered list of tracks. A track is a named set of markers, which can specify a frame rate for all markers in the set. See also xmpDM:markers.
xmpDM:videoAlphaMode	closed Choice of Text	External	The alpha mode. One of: straight pre-multiplied

Property	Value type	Category	Description
xmpDM: videoAlphaPremultipleColor	Colorant	External	A color in CMYK or RGB to be used as the pre-multiple color when alpha mode is `pre-multiplied`.
xmpDM: videoAlphaUnityIsTransparent	Boolean	Internal	When `true`, unity is clear, when `false`, it is opaque.
xmpDM:videoColorSpace	closed Choice of Text	Internal	The color space. One of: `sRGB` (used by Photoshop) `CCIR-601` (used for NTSC) `CCIR-709` (used for HD)
xmpDM:videoCompressor	Text	Internal	Video compression used. For example, `jpeg`.

xmpDM:videoFieldOrder	closed Choice of Text	Internal	The field order for video. One of:
			`Upper`
			`Lower`
			`Progressive`

xmpDM:videoFrameRate	open Choice of Text	Internal	The video frame rate. One of:
			`24`
			`NTSC`
			`PAL`

xmpDM:videoFrameSize	Dimensions	Internal	The frame size. For example: `w:720, h: 480, unit:pixels`
xmpDM:videoModDate	Date	Internal	The date and time when the video was last modified.

xmpDM:videoPixelDepth	closed Choice of Text	Internal	The size in bits of each color component of a pixel. Standard Windows 32-bit pixels have 8 bits per component. One of:
			`8Int`
			`16Int`
			`32Int`
			`32Float`

xmpDM:videoPixelAspectRatio	Rational	Internal	The aspect ratio, expressed as wd/ht. For example: `"648/720"` = 0.9

3 Specialized Schemas

This chapter contains schema definitions for schemas that are specialized for Adobe applications and usages. The following schemas definitions are included here:

Adobe PDF schema

This schema specifies properties used with Adobe PDF documents.

- The schema namespace URI is `http://ns.adobe.com/pdf/1.3/`
- The preferred schema namespace prefix is `pdf`

Property	Value type	Category	Description
`pdf:Keywords`	Text	External	Keywords.
`pdf:PDFVersion`	Text	Internal	The PDF file version (for example: 1.0, 1.3, and so on).
`pdf:Producer`	AgentName	Internal	The name of the tool that created the PDF document.

Photoshop schema

This schema specifies properties used by Adobe Photoshop.

- The schema namespace URI is `http://ns.adobe.com/photoshop/1.0/`
- The preferred schema namespace prefix is `photoshop`

Property	Value type	Category	Description
`photoshop:AuthorsPosition`	Text	External	By-line title.
`photoshop:CaptionWriter`	ProperName	External	Writer/editor.
`photoshop:Category`	Text	External	Category. Limited to 3 7-bit ASCII characters.
`photoshop:City`	Text	External	City.
`photoshop:Country`	Text	External	Country/primary location.
`photoshop:Credit`	Text	External	Credit.

Property	Value type	Category	Description
photoshop:DateCreated	Date	External	The date the intellectual content of the document was created (rather than the creation date of the physical representation), following IIM conventions. For example, a photo taken during the American Civil War would have a creation date during that epoch (1861-1865) rather than the date the photo was digitized for archiving.
photoshop:Headline	Text	External	Headline.
photoshop:Instructions	Text	External	Special instructions.
photoshop:Source	Text	External	Source.
photoshop:State	Text	External	Province/state.
photoshop:SupplementalCategories	bag Text	External	Supplemental category.
photoshop:TransmissionReference	Text	External	Original transmission reference.
photoshop:Urgency	Integer	External	Urgency. Valid range is 1-8.

Camera Raw schema

This schema specifies settings associated with image files produced in camera raw mode.

- The schema namespace URI is `http://ns.adobe.com/camera-raw-settings/1.0/`
- The preferred schema namespace prefix is `crs`

Property	Value type	Category	Description
`crs:AutoBrightness`	Boolean	External	When true, "Brightness" is automatically adjusted.
`crs:AutoContrast`	Boolean	External	When true, "Contrast" is automatically adjusted.
`crs:AutoExposure`	Boolean	External	When true, "Exposure" is automatically adjusted.
`crs:AutoShadows`	Boolean	External	When true, "Shadows" is automatically adjusted.
`crs:BlueHue`	Integer	External	"Blue Hue" setting. Range -100 to 100.
`crs:BlueSaturation`	Integer	External	"Blue Saturation" setting. Range -100 to 100.
`crs:Brightness`	Integer	External	"Brightness" setting. Range 0 to 150.
`crs:CameraProfile`	Text	External	"Camera Profile" setting.
`crs:ChromaticAberrationB`	Integer	External	"Chromatic Aberration, Fix Blue/Yellow Fringe" setting. Range -100 to 100.
`crs:ChromaticAberrationR`	Integer	External	"Chromatic Aberration, Fix Red/Cyan Fringe" setting. Range -100 to 100.
`crs:ColorNoiseReduction`	Integer	External	"Color Noise Reduction" setting. Range 0 to 100.
`crs:Contrast`	Integer	External	"Contrast" setting. Range -50 to 100.
`crs:CropTop`	Real	External	When `HasCrop` is `true`, top of crop rectangle
`crs:CropLeft`	Real	External	When `HasCrop` is `true`, left of crop rectangle.
`crs:CropBottom`	Real	External	When `HasCrop` is `true`, bottom of crop rectangle.
`crs:CropRight`	Real	External	When `HasCrop` is `true`, right of crop rectangle.
`crs:CropAngle`	Real	External	When `HasCrop` is `true`, angle of crop rectangle.
`crs:CropWidth`	Real	External	Width of resulting cropped image in `CropUnits` units.
`crs:CropHeight`	Real	External	Height of resulting cropped image in `CropUnits` units.

Property	Value type	Category	Description
crs:CropUnits	Integer	External	Units for CropWidth and CropHeight. One of: 0 = pixels 1 = inches 2 = cm
crs:Exposure	Real	External	"Exposure" setting. Range -4.0 to 4.0.
crs:GreenHue	Integer	External	"Green Hue" setting. Range -100 to 100.
crs:GreenSaturation	Integer	External	"Green Saturation" setting. Range -100 to 100.
crs:HasCrop	Boolean	External	When true, image has a cropping rectangle.
crs:HasSettings	Boolean	External	When true, non-default camera raw settings.
crs:LuminanceSmoothing	Integer	External	"Luminance Smoothing" setting. Range 0 to 100.
crs:RawFileName	Text	Internal	File name for raw file (not a complete path).
crs:RedHue	Integer	External	"Red Hue" setting. Range -100 to 100.
crs:RedSaturation	Integer	External	"Red Saturation" setting. Range -100 to 100.
crs:Saturation	Integer	External	"Saturation" setting. Range -100 to 100.
crs:Shadows	Integer	External	"Shadows" setting. Range 0 to 100.
crs:ShadowTint	Integer	External	"Shadow Tint" setting. Range -100 to 100.
crs:Sharpness	Integer	External	"Sharpness" setting. Range 0 to 100.
crs:Temperature	Integer	External	"Temperature" setting. Range 2000 to 50000.
crs:Tint	Integer	External	"Tint" setting. Range -150 to 150.
crs:ToneCurve	Seq of points (Integer, Integer)	External	Array of points (Integer, Integer) defining a "Tone Curve."
crs:ToneCurveName	Choice Text	Internal	The name of the Tone Curve described by ToneCurve. One of: Linear Medium Contrast Strong Contrast Custom or a user-defined preset name
crs:Version	Text	Internal	Version of Camera Raw plug-in.
crs:VignetteAmount	Integer	External	"Vignetting Amount" setting. Range -100 to 100.

Property	Value type	Category	Description
`crs:VignetteMidpoint`	Integer	External	"Vignetting Midpoint" setting. Range 0 to 100.

`crs:WhiteBalance`	Closed Choice of Text	External	"White Balance" setting. One of:
			`As Shot`
			`Auto`
			`Daylight`
			`Cloudy`
			`Shade`
			`Tungsten`
			`Fluorescent`
			`Flash`
			`Custom`

EXIF schemas

EXIF is a metadata standard for image files, used widely by digital cameras. The EXIF 2.2 specification can be found at http://www.exif.org/specifications.html.

There are two XMP schemas that correspond to parts of the EXIF 2.2 specification, described in the following sections:

- "EXIF schema for TIFF properties" on page 48
- "EXIF schema for EXIF-specific properties" on page 50
- "EXIF schema for additional EXIF properties" on page 58 describes a namespace that defines additional properties for the equipment used to produce EXIF data.

The property descriptions assume that the reader has some familiarity with EXIF metadata. The XMP property names are identical to the names used within the EXIF specification; more complete descriptions of the properties can be found in the specification.

The following sections provide further information:

- "Data representation and conversion" on page 58 describes guidelines for converting between the XMP and EXIF formats, with examples.
- "EXIF schema value types" on page 23 describes EXIF-specific value types.

  NOTE: XMP properties of type Date include fractional seconds; therefore EXIF properties for fractional seconds (`SubSecTime`, `SubSecTimeOriginal`, `SubSecTimeDigitized`) are included in the "main XMP property" list.

EXIF schema for TIFF properties

The following table lists the properties for TIFF-derived data. Only those TIFF properties that are mentioned in the EXIF 2.2 specification are included here.

- The schema name is `http://ns.adobe.com/tiff/1.0/`
- The preferred schema namespace prefix is `tiff`

Property	Value type	Category	Description
`tiff:ImageWidth`	Integer	Internal	TIFF tag 256, 0x100. Image width in pixels.
`tiff:ImageLength`	Integer	Internal	TIFF tag 257, 0x101. Image height in pixels.
`tiff:BitsPerSample`	seq Integer	Internal	TIFF tag 258, 0x102. Number of bits per component in each channel.
`tiff:Compression`	Closed Choice of Integer	Internal	TIFF tag 259, 0x103. Compression scheme:1 = uncompressed; 6 = JPEG.
`tiff:PhotometricInterpretation`	Closed Choice of Integer	Internal	TIFF tag 262, 0x106. Pixel Composition: 2 = RGB; 6 = YCbCr.

Property	Value type	Category	Description
tiff:Orientation	Closed Choice of Integer	Internal	TIFF tag 274, 0x112. Orientation: 1 = 0th row at top, 0th column at left 2 = 0th row at top, 0th column at right 3 = 0th row at bottom, 0th column at right 4 = 0th row at bottom, 0th column at left 5 = 0th row at left, 0th column at top 6 = 0th row at right, 0th column at top 7 = 0th row at right, 0th column at bottom 8 = 0th row at left, 0th column at bottom
tiff:SamplesPerPixel	Integer	Internal	TIFF tag 277, 0x115. Number of components per pixel.
tiff:PlanarConfiguration	Closed Choice of Integer	Internal	TIFF tag 284, 0x11C. Data layout 1 = chunky 2 = planar
tiff:YCbCrSubSampling	Closed Choice of seq Integer	Internal	TIFF tag 530, 0x212. Sampling ratio of chrominance components: [2, 1] = YCbCr4:2:2 [2, 2] = YCbCr4:2:0
tiff:YCbCrPositioning	Closed Choice of Integer	Internal	TIFF tag 531, 0x213. Position of chrominance vs. luminance components: 1 = centered 2 = co-sited
tiff:XResolution	Rational	Internal	TIFF tag 282, 0x11A. Horizontal resolution in pixels per unit.
tiff:YResolution	Rational	Internal	TIFF tag 283, 0x11B. Vertical resolution in pixels per unit.
tiff:ResolutionUnit	Closed Choice of Integer	Internal	TIFF tag 296, 0x128. Unit used for XResolution and YResolution. Value is one of: 2 = inches 3 = centimeters
tiff:TransferFunction	seq Integer	Internal	TIFF tag 301, 0x12D. Transfer function for image described in tabular style with 3 * 256 entries.
tiff:WhitePoint	seq Rational	Internal	TIFF tag 318, 0x13E. Chromaticity of white point.
tiff:PrimaryChromaticities	seq Rational	Internal	TIFF tag 319, 0x13F. Chromaticity of the three primary colors.
tiff:YCbCrCoefficients	seq Rational	Internal	TIFF tag 529, 0x211. Matrix coefficients for RGB to YCbCr transformation.
tiff:ReferenceBlackWhite	seq Rational	Internal	TIFF tag 532, 0x214. Reference black and white point values.

Property	Value type	Category	Description
tiff:DateTime	Date	Internal	TIFF tag 306, 0x132 (primary) and EXIF tag 37520, 0x9290 (subseconds). Date and time of image creation (no time zone in EXIF), stored in ISO 8601 format, not the original EXIF format. This property includes the value for the EXIF SubSecTime attribute. This property is stored in XMP as xmp:ModifyDate.
tiff:ImageDescription	Lang Alt	External	TIFF tag 270, 0x10E. Description of the image. This property is stored in XMP as dc:description.
tiff:Make	ProperName	Internal	TIFF tag 271, 0x10F. Manufacturer of recording equipment.
tiff:Model	ProperName	Internal	TIFF tag 272, 0x110. Model name or number of equipment.
tiff:Software	AgentName	Internal	TIFF tag 305, 0x131. Software or firmware used to generate image. This property is stored in XMP as xmp:CreatorTool.
tiff:Artist	ProperName	External	TIFF tag 315, 0x13B. Camera owner, photographer or image creator. This property is stored in XMP as the first item in the dc:creator array.
tiff:Copyright	Lang Alt	External	TIFF tag 33432, 0x8298. Copyright information. This property is stored in XMP as dc:rights.

EXIF schema for EXIF-specific properties

The following table lists the properties defined solely by EXIF.

NOTES: A number of EXIF 2.2 properties are not included in XMP. These are generally properties that relate directly to the image stream, or that are of little use without access to the image stream. A general XMP principle is that XMP metadata should have value in and of itself, separate from the primary file content. The omitted properties include: StripOffsets, RowsPerStrip, StripByteCounts, JPEGInterchangeFormat, and JPEGInterchangeFormatLength

Properties beginning with "GPS" are GPS properties that are also used by DIG-35 and are part of the JPEG-2000 standard.

- The schema name is http://ns.adobe.com/exif/1.0/
- The preferred schema namespace prefix is exif

Property	Value type	Category	Description
exif:ExifVersion	Closed Choice of Text	Internal	EXIF tag 36864, 0x9000. EXIF version number.
exif:FlashpixVersion	Closed Choice of Text	Internal	EXIF tag 40960, 0xA000. Version of FlashPix.
exif:ColorSpace	Closed Choice of Integer	Internal	EXIF tag 40961, 0xA001. Color space information: 1 = sRGB 65535 = uncalibrated
exif:ComponentsConfiguration	Closed Choice of seq Integer	Internal	EXIF tag 37121, 0x9101. Configuration of components in data: 4 5 6 0 (if RGB compressed data), 1 2 3 0 (other cases). 0 = does not exist 1 = Y 2 = Cb 3 = Cr 4 = R 5 = G 6 = B
exif:CompressedBitsPerPixel	Rational	Internal	EXIF tag 37122, 0x9102. Compression mode used for a compressed image is indicated in unit bits per pixel.
exif:PixelXDimension	Integer	Internal	EXIF tag 40962, 0xA002. Valid image width, in pixels.
exif:PixelYDimension	Integer	Internal	EXIF tag 40963, 0xA003. Valid image height, in pixels.
exif:UserComment	Lang Alt	External	EXIF tag 37510, 0x9286. Comments from user.
exif:RelatedSoundFile	Text	Internal	EXIF tag 40964, 0xA004. An “8.3” file name for the related sound file.
exif:DateTimeOriginal	Date	Internal	EXIF tags 36867, 0x9003 (primary) and 37521, 0x9291 (subseconds). Date and time when original image was generated, in ISO 8601 format. Includes the EXIF `SubSecTimeOriginal` data. Note that EXIF date-time values have no time zone information.
exif:DateTimeDigitized	Date	Internal	EXIF tag 36868, 0x9004 (primary) and 37522, 0x9292 (subseconds). Date and time when image was stored as digital data, can be the same as `DateTimeOriginal` if originally stored in digital form. Stored in ISO 8601 format. Includes the EXIF `SubSecTimeDigitized` data.
exif:ExposureTime	Rational	Internal	EXIF tag 33434, 0x829A. Exposure time in seconds.
exif:FNumber	Rational	Internal	EXIF tag 33437, 0x829D. F number.

Property	Value type	Category	Description
exif:ExposureProgram	Closed Choice of Integer	Internal	EXIF tag 34850, 0x8822. Class of program used for exposure: 0 = not defined 1 = Manual 2 = Normal program 3 = Aperture priority 4 = Shutter priority 5 = Creative program 6 = Action program 7 = Portrait mode 8 = Landscape mode
exif:SpectralSensitivity	Text	Internal	EXIF tag 34852, 0x8824. Spectral sensitivity of each channel.
exif:ISOSpeedRatings	seq Integer	Internal	EXIF tag 34855, 0x8827. ISO Speed and ISO Latitude of the input device as specified in ISO 12232.
exif:OECF	OECF/SFR	Internal	EXIF tag 34856, 0x8828. Opto-Electronic Conversion Function as specified in ISO 14524.
exif:ShutterSpeedValue	Rational	Internal	EXIF tag 37377, 0x9201. Shutter speed, unit is APEX. See Annex C of the EXIF specification.
exif:ApertureValue	Rational	Internal	EXIF tag 37378, 0x9202. Lens aperture, unit is APEX.
exif:BrightnessValue	Rational	Internal	EXIF tag 37379, 0x9203. Brightness, unit is APEX.
exif:ExposureBiasValue	Rational	Internal	EXIF tag 37380, 0x9204. Exposure bias, unit is APEX.
exif:MaxApertureValue	Rational	Internal	EXIF tag 37381, 0x9205. Smallest F number of lens, in APEX.
exif:SubjectDistance	Rational	Internal	EXIF tag 37382, 0x9206. Distance to subject, in meters.
exif:MeteringMode	Closed Choice of Integer	Internal	EXIF tag 37383, 0x9207. Metering mode: 0 = unknown 1 = Average 2 = CenterWeightedAverage 3 = Spot 4 = MultiSpot 5 = Pattern 6 = Partial 255 = other

Property	Value type	Category	Description
exif:LightSource	Closed Choice of Integer	Internal	EXIF tag 37384, 0x9208. EXIF tag, 0x. Light source: 0 = unknown 1 = Daylight 2 = Fluorescent 3 = Tungsten 4 = Flash 9 = Fine weather 10 = Cloudy weather 11 = Shade 12 = Daylight fluorescent (D 5700 – 7100K) 13 = Day white fluorescent (N 4600 – 5400K) 14 = Cool white fluorescent (W 3900 – 4500K) 15 = White fluorescent (WW 3200 – 3700K) 17 = Standard light A 18 = Standard light B 19 = Standard light C 20 = D55 21 = D65 22 = D75 23 = D50 24 = ISO studio tungsten 255 = other
exif:Flash	Flash	Internal	EXIF tag 37385, 0x9209. Strobe light (flash) source data.
exif:FocalLength	Rational	Internal	EXIF tag 37386, 0x920A. Focal length of the lens, in millimeters.
exif:SubjectArea	seq Integer	Internal	EXIF tag 37396, 0x9214. The location and area of the main subject in the overall scene.
exif:FlashEnergy	Rational	Internal	EXIF tag 41483, 0xA20B. Strobe energy during image capture.
exif:SpatialFrequencyResponse	OECF/SFR	Internal	EXIF tag 41484, 0xA20C. Input device spatial frequency table and SFR values as specified in ISO 12233.
exif:FocalPlaneXResolution	Rational	Internal	EXIF tag 41486, 0xA20E. Horizontal focal resolution, measured pixels per unit.
exif:FocalPlaneYResolution	Rational	Internal	EXIF tag 41487, 0xA20F. Vertical focal resolution, measured in pixels per unit.
exif:FocalPlaneResolutionUnit	Closed Choice of Integer	Internal	EXIF tag 41488, 0xA210. Unit used for FocalPlaneXResolution and FocalPlaneYResolution. 2 = inches 3 = centimeters

Property	Value type	Category	Description
exif:SubjectLocation	seq Integer	Internal	EXIF tag 41492, 0xA214. Location of the main subject of the scene. The first value is the horizontal pixel and the second value is the vertical pixel at which the main subject appears.
exif:ExposureIndex	Rational	Internal	EXIF tag 41493, 0xA215. Exposure index of input device.
exif:SensingMethod	Closed Choice of Integer	Internal	EXIF tag 41495, 0xA217. Image sensor type on input device: 1 = Not defined 2 = One-chip color area sensor 3 = Two-chip color area sensor 4 = Three-chip color area sensor 5 = Color sequential area sensor 7 = Trilinear sensor 8 = Color sequential linear sensor
exif:FileSource	Closed Choice of Integer	Internal	EXIF tag 41728, 0xA300. Indicates image source: 3 (DSC) is the only choice.
exif:SceneType	Closed Choice of Integer	Internal	EXIF tag 41729, 0xA301. Indicates the type of scene: 1 (directly photographed image) is the only choice.
exif:CFAPattern	CFAPattern	Internal	EXIF tag 41730, 0xA302. Color filter array geometric pattern of the image sense.
exif:CustomRendered	Closed Choice of Integer	Internal	EXIF tag 41985, 0xA401. Indicates the use of special processing on image data: 0 = Normal process 1 = Custom process
exif:ExposureMode	Closed Choice of Integer	Internal	EXIF tag 41986, 0xA402. Indicates the exposure mode set when the image was shot: 0 = Auto exposure 1 = Manual exposure 2 = Auto bracket
exif:WhiteBalance	Closed Choice of Integer	Internal	EXIF tag 41987, 0xA403. Indicates the white balance mode set when the image was shot: 0 = Auto white balance 1 = Manual white balance
exif:DigitalZoomRatio	Rational	Internal	EXIF tag 41988, 0xA404. Indicates the digital zoom ratio when the image was shot.
exif:FocalLengthIn35mmFilm	Integer	Internal	EXIF tag 41989, 0xA405. Indicates the equivalent focal length assuming a 35mm film camera, in mm. A value of 0 means the focal length is unknown. Note that this tag differs from the FocalLength tag.

Property	Value type	Category	Description
exif:SceneCaptureType	Closed Choice of Integer	Internal	EXIF tag 41990, 0xA406. Indicates the type of scene that was shot: 0 = Standard 1 = Landscape 2 = Portrait 3 = Night scene
exif:GainControl	Closed Choice of Integer	Internal	EXIF tag 41991, 0xA407. Indicates the degree of overall image gain adjustment: 0 = None 1 = Low gain up 2 = High gain up 3 = Low gain down 4 = High gain down
exif:Contrast	Closed Choice of Integer	Internal	EXIF tag 41992, 0xA408. Indicates the direction of contrast processing applied by the camera: 0 = Normal 1 = Soft 2 = Hard
exif:Saturation	Closed Choice of Integer	Internal	EXIF tag 41993, 0xA409. Indicates the direction of saturation processing applied by the camera: 0 = Normal 1 = Low saturation 2 = High saturation
exif:Sharpness	Closed Choice of Integer	Internal	EXIF tag 41994, 0xA40A. Indicates the direction of sharpness processing applied by the camera: 0 = Normal 1 = Soft 2 = Hard
exif:DeviceSettingDescription	DeviceSettings	Internal	EXIF tag 41995, 0xA40B. Indicates information on the picture-taking conditions of a particular camera model.
exif:SubjectDistanceRange	Closed Choice of Integer	Internal	EXIF tag 41996, 0xA40C. Indicates the distance to the subject: 0 = Unknown 1 = Macro 2 = Close view 3 = Distant view
exif:ImageUniqueID	Text	Internal	EXIF tag 42016, 0xA420. An identifier assigned uniquely to each image. It is recorded as a 32 character ASCII string, equivalent to hexadecimal notation and 128-bit fixed length.

Property	Value type	Category	Description
`exif:GPSVersionID`	Text	Internal	GPS tag 0, 0x00. A decimal encoding of each of the four EXIF bytes with period separators. The current value is "2.0.0.0".
`exif:GPSLatitude`	GPSCoordinate	Internal	GPS tag 2, 0x02 (position) and 1, 0x01 (North/South). Indicates latitude.
`exif:GPSLongitude`	GPSCoordinate	Internal	GPS tag 4, 0x04 (position) and 3, 0x03 (East/West). Indicates longitude.
`exif:GPSAltitudeRef`	Closed Choice of Integer	Internal	GPS tag 5, 0x5. Indicates whether the altitude is above or below sea level: 0 = Above sea level 1 = Below sea level
`exif:GPSAltitude`	Rational	Internal	GPS tag 6, 0x06. Indicates altitude in meters.
`exif:GPSTimeStamp`	Date	Internal	GPS tag 29 (date), 0x1D, and, and GPS tag 7 (time), 0x07. Time stamp of GPS data, in Coordinated Universal Time. The `GPSDateStamp` tag is new in EXIF 2.2. The GPS timestamp in EXIF 2.1 does not include a date. If not present, the date component for the XMP should be taken from `exif:DateTimeOriginal`, or if that is also lacking from `exif:DateTimeDigitized`. If no date is available, do not write `exif:GPSTimeStamp` to XMP.
`exif:GPSSatellites`	Text	Internal	GPS tag 8, 0x08. Satellite information, format is unspecified.
`exif:GPSStatus`	Closed Choice of Text	Internal	GPS tag 9, 0x09. Status of GPS receiver at image creation time: A = measurement in progress V = measurement is interoperability
`exif:GPSMeasureMode`	Text	Internal	GPS tag 10, 0x0A. GPS measurement mode, Text type: 2 = two-dimensional measurement 3 = three-dimensional measurement
`exif:GPSDOP`	Rational	Internal	GPS tag 11, 0x0B. Degree of precision for GPS data.
`exif:GPSSpeedRef`	Closed Choice of Text	Internal	GPS tag 12, 0x0C. Units used to speed measurement: K = kilometers per hour M = miles per hour N = knots
`exif:GPSSpeed`	Rational	Internal	GPS tag 13, 0x0D. Speed of GPS receiver movement.

Property	Value type	Category	Description
exif:GPSTrackRef	Closed Choice of Text	Internal	GPS tag 14, 0x0E. Reference for movement direction: T = true direction M = magnetic direction
exif:GPSTrack	Rational	Internal	GPS tag 15, 0x0F. Direction of GPS movement, values range from 0 to 359.99.
exif:GPSImgDirectionRef	Closed Choice of Text	Internal	GPS tag 16, 0x10. Reference for movement direction: T = true direction M = magnetic direction
exif:GPSImgDirection	Rational	Internal	GPS tag 17, 0x11. Direction of image when captured, values range from 0 to 359.99.
exif:GPSMapDatum	Text	Internal	GPS tag 18, 0x12. Geodetic survey data.
exif:GPSDestLatitude	GPSCoordinate	Internal	GPS tag 20, 0x14 (position) and 19, 0x13 (North/South). Indicates destination latitude.
exif:GPSDestLongitude	GPSCoordinate	Internal	GPS tag 22, 0x16 (position) and 21, 0x15 (East/West). Indicates destination longitude.
exif:GPSDestBearingRef	Closed Choice of Text	Internal	GPS tag 23, 0x17. Reference for movement direction: T = true direction M = magnetic direction
exif:GPSDestBearing	Rational	Internal	GPS tag 24, 0x18. Destination bearing, values from 0 to 359.99.
exif:GPSDestDistanceRef	Closed Choice of Text	Internal	GPS tag 25, 0x19. Units used for speed measurement: K = kilometers M = miles N = knots
exif:GPSDestDistance	Rational	Internal	GPS tag 26, 0x1A. Distance to destination.
exif:GPSProcessingMethod	Text	Internal	GPS tag 27, 0x1B. A character string recording the name of the method used for location finding.
exif:GPSAreaInformation	Text	Internal	GPS tag 28, 0x1C. A character string recording the name of the GPS area.
exif:GPSDifferential	Closed Choice of Integer	Internal	GPS tag 30, 0x1E. Indicates whether differential correction is applied to the GPS receiver: 0 = Without correction 1 = Correction applied

EXIF schema for additional EXIF properties

The following table lists additional properties that describe the equipment used to produce EXIF data.

- The schema name is `http://ns.adobe.com/exif/1.0/aux/`
- The preferred schema namespace prefix is `aux`

Property	Value type	Category	Description
`aux:Lens`	Text	Internal	A description of the lens used to take the photograph. For example, "70-200 mm f/2.8-4.0".
`aux:SerialNumber`	Text	Internal	The serial number of the camera or camera body used to take the photograph.

Data representation and conversion

This section describes the mapping from the native EXIF 2.2 metadata format to the XMP format. It explains how to do the conversion without losing significant data, and describes the resulting XMP representation.

> **NOTE:** If a particular tag is omitted from an EXIF file, the corresponding XMP property must also be omitted. An XMP property must not be created based on the default value of a missing EXIF tag.

The EXIF to XMP type mappings are designed to be lossless in most cases. The main issues are for EXIF text values. When converting from XMP, integers that are specified optionally as `short` or `long` in EXIF should be represented as `short` if the value is in the range –32768 to +32767, otherwise they should be `long`.

EXIF text

EXIF text values are a sequence of ASCII characters with a null terminator; XMP text values are Unicode characters in UTF-8 with no null terminator. When converting EXIF to XMP, the null terminator is dropped; the remaining ASCII codes are legitimate UTF-8 values. When converting from XMP to EXIF, non-ASCII characters are escaped (using URL escaping as specified in http://www.w3.org/Addressing/URL/4_Recommentations.html); ASCII characters in the range of 0 through 127 are not escaped (for example, spaces); and a null terminator is added.

XMP text values can be localized. For properties of type Lang Alt, an array of localized text values can be supplied. When converting from EXIF to XMP, the value supplied by the EXIF metadata should be written to the default entry (`[@xml:lang='x-default']`). When converting from XMP to EXIF, the default entry should be used to supply the EXIF metadata.

EXIF dates

All date/time values are stored in XMP using ISO 8601 format. This is a combined date and time, with fractional seconds, and a time zone designation. The binary EXIF values generally separate the fractional seconds. EXIF 2.1 lacks time zone information; this has been partially added in EXIF 2.2. When converting to XMP, the fractional seconds should be included. EXIF date-time values have no time zone information, so any UTC offsets should be ignored when reading the EXIF value. Convert to XMP assuming a local time.

Example

The following is an example of EXIF 2.2 metadata and the corresponding XMP metadata as it might be converted from the EXIF data.

The EXIF data

```
IFD 0 [1]
Make = "Canon"
Model = "Canon PowerShot S300"
Orientation = "1"
XResolution = "180/1" (180.00)
YResolution = "180/1" (180.00)
ResolutionUnit = "2"
DateTime = "2001:07:25 20:18:27"
YCbCrPositioning = "1"
ExposureTime = "1/60" (0.0167)
FNumber = "27/10" (2.70)
ExifVersion = "30 32 31 30"
DateTimeOriginal = "2001:07:25 20:18:27"
DateTimeDigitized = "2001:07:25 20:18:27"
ComponentsConfiguration = "1 2 3 0"
CompressedBitsPerPixel = "3/1" (3.00)
ShutterSpeedValue = "189/32" (5.91)
ApertureValue = "93/32" (2.91)
ExposureBiasValue = "0/3" (0.00)
MaxApertureValue = "187820/65536" (2.8659)
SubjectDistance = "913/1000" (0.9130)
MeteringMode = "5"
Flash = "0x01"
FocalLength = "173/32" (5.41)
```

The XMP metadata

This example uses the RDF shorthand notation of representing simple properties as XML attributes instead of XML elements.

```
<rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
    <rdf:Description about='' xmlns:tiff='http://ns.adobe.com/tiff/1.0'
       tiff:Make='Canon'
       tiff:Model='Canon PowerShot S300'
       tiff:Orientation='1'
       tiff:XResolution='180/1'
       tiff:YResolution='180/1'
       tiff:ResolutionUnit='2'
       tiff:DateTime='2001-07-25T20:18:27-07:00'
       tiff:YCbCrPositioning='1'>
    </rdf:Description>
```

```
    <rdf:Description about='' xmlns:exif='http://ns.adobe.com/exif/1.0'
       exif:ExposureTime='1/60'
       exif:FNumber='27/10'
       exif:ExifVersion='0210'
       exif:DateTimeOriginal='2001-07-25T20:18:27-07:00'
       exif:DateTimeDigitized='2001-07-25T20:18:27-07:00'
       exif:CompressedBitsPerPixel='3/1'
       exif:ShutterSpeedValue='189/32'
       exif:ApertureValue='93/32'
       exif:ExposureBiasValue='0/3'
       exif:MaxApertureValue='187820/65536'
       exif:SubjectDistance='913/1000'

       exif:MeteringMode='5'
       exif:Flash='1'
       exif:FocalLength='173/32'>
       <exif:ComponentsConfiguration>
          <rdf:Seq>
             <rdf:li>1</rdf:li>
             <rdf:li>2</rdf:li>
             <rdf:li>3</rdf:li>
             <rdf:li>0</rdf:li>
          </rdf:Seq>
       </exif:ComponentsConfiguration>
    </rdf:Description>

</rdf:RDF>
```