GUIDELINES FOR HANDLING IMAGE METADATA

Version 2.0

November 2010

METADATA WORKING GROUP

www.metadataworkinggroup.org

Copyrights

Terms and Conditions

This document is made available by Adobe Systems Inc., Apple Inc., Canon Inc., Microsoft Corp., Nokia Corp. and Sony Corp. (collectively, the "Authors") and grants you (either an individual or an entity) and your affiliates ("Licensee") this license. Licensee agrees that Licensee has read, understood and will comply with these terms and conditions.

1. Definitions

1.1 "*Licensed Products*" means only those specific portions of products (hardware, software or combinations thereof) that implement and are compliant with all Normative Portions of the Guidelines.

1.2 "*Normative Portions*" means a portion of the Guidelines that must be implemented to comply with such guidelines. If such guidelines define optional parts, Normative Portions include those portions of the optional part that must be implemented if the implementation is to comply with such optional part.

1.3 "*Necessary Claims*" are those claims of a patent or patent application, throughout the world, excluding design patents and design registrations, owned or controlled, or that can be sublicensed in compliance with the requirements of this Agreement, by the party or its affiliates now or at any future time and which would necessarily be infringed by implementation of the Guidelines. A claim is necessarily infringed hereunder only when it is not possible to avoid infringing it because there is no non-infringing alternative for implementing the Normative Portions of the Guidelines. Notwithstanding the foregoing, Necessary Claims shall not include any claims other than as set forth above even if contained in the same patent as Necessary Claims; or that read solely on any implementations of any portion of the Guidelines that are not required by the Normative Portions of the Guidelines, or that, if licensed, would require a payment of royalties by the licensor to unaffiliated third parties. Moreover, Necessary Claims shall not include (i) any enabling technologies that may be necessary to make or use any Licensed Product but are not themselves expressly set forth in the Guidelines (e.g., semiconductor manufacturing technology, compiler technology, object oriented technology, basic operating system technology, data and voice networking technology, and the like); or (ii) the implementation of other published standards developed elsewhere and merely referred to in the body of the Guidelines, or (iii) any Licensed Product and any combinations thereof the purpose or function of which is not required for compliance with the Guidelines. For purposes of this definition, the Guidelines shall be deemed to include only architectural and interconnection requirements essential for interoperability and shall not include any implementation examples unless such implementation examples are expressly identified as being required for compliance with the Guidelines.

1.4 "*Guidelines*" mean the Guidelines For Handling Image Metadata Version 2.0.

2. License

2.1 Copyright: Licensee must include the following on ALL copies of the Guidelines, or portions thereof that Licensee makes:

2.1.1. A link or URL to the Guidelines at this location: http://www.metadataworkinggroup.org

2.1.2. The copyright notice as shown in the Guidelines.

2.2 Patent: The Authors each will grant Licensee a royalty-free license under reasonable, non-discriminatory terms and conditions to their Necessary Claims to make, have made, use, reproduce market, import, offer to sell and sell, and to otherwise distribute Licensed Products. Licensee agrees to grant Authors and their affiliates a royalty-free license under reasonable, non-discriminatory terms and conditions to its Necessary Claims to make, have made, use, reproduce market, import, offer to sell and sell, and to otherwise distribute Licensed Products. Nothing herein shall prevent the Authors from charging a reasonable royalty for such Necessary Claims to any party who is offering their Necessary Claims on royalty bearing terms.

3. Limitations

3.1 No Warranty: THE “GUIDELINES FOR HANDLING IMAGE METADATA VERSION 2.0” GUIDELINES IS PROVIDED “AS IS,” AND THE AUTHORS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE GUIDELINES ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

3.2 No Liability: THE AUTHORS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THE GUIDELINES.

3.3 Trademark: The name and trademarks of the Authors may NOT be used in any manner, including advertising or publicity pertaining to the Guidelines or its contents without specific, written prior permission. Title to copyright in the Guidelines will at all times remain with the Authors.

3.4 No Other Rights: No other rights are granted by implication, estoppel or otherwise.

Normative Sections

This document attempts to conform to the keyword usage practices defined in RFC 2119. This RFC defines the use and strength of the capitalized terms MUST, MUST NOT, SHOULD, SHOULD NOT and MAY. All sections and appendixes, except the first chapter “Introduction”, are normative, unless they are explicitly indicated to be informative.

These imperatives are used to highlight those requirements that are required to insure interoperability and drive compatibility.

References

This document includes the following references to third party documents:

Metadata Specifications

Exif 2.3

http://www.cipa.jp/english/hyoujunka/kikaku/pdf/DC-008-2010_E.pdf

DCF 2.0 (2010)

http://www.cipa.jp/english/hyoujunka/kikaku/pdf/DC-009-2010_E.pdf

IPTC-IIM 4.1

http://www.iptc.org/std/IIM/4.1/specification/IIMV4.1.pdf

IPTC Core 1.0

http://www.iptc.org/std/Iptc4xmpCore/1.0/specification/Iptc4xmpCore_1.0-spec-XMPSchema_8.pdf

IPTC Core 1.1 & IPTC Extension 1.1

http://www.iptc.org/std/photometadata/specification/IPTC-PhotoMetadata-201007.pdf

XMP

http://www.adobe.com/devnet/xmp/

File Format Specifications

JPEG

http://www.jpeg.org/jpeg/

TIFF

http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf

PSD/PSIRs

http://www.adobe.com/go/psir

Miscellaneous

RDF

http://www.w3.org/TR/rdf-schema

Dublin Core

http://dublincore.org/documents/dces

RFC2119

http://www.ietf.org/rfc/rfc2119.txt

Date and Time (W3C)

http://www.w3.org/TR/NOTE-datetime

TABLE OF CONTENTS

1. ABOUT THIS DOCUMENT

The “Guidelines for Handling Image Metadata” document is being updated regularly to incorporate user feedback as well as addressing new areas that have been identified as valuable in the scope of the Metadata Working Group.

This document can be found at the following address: http://www.metadataworkinggroup.org

Document history

Version	Date	Description
1.0	September 2008	Initial guidance document of Metadata Working Group
1.0.1	February 2009	First update incorporating users’ feedback
2.0	October 2010	Major update and addition of new schemata for image region metadata, hierarchical keywords and collections

New in this version

- Updated document structure
- Clarified text encoding section
- Provided additional guidance for location based metadata
- Added schema support for the following consumer metadata:
  - Image Region Metadata
  - Hierarchical Keywords
  - Collections

Acknowledgments

This document is the result of a collaborative effort amongst Metadata Working Group members. We wish to thank all of those members of the metadata community who spent time and energy to review this document and provide us with valuable feedback.

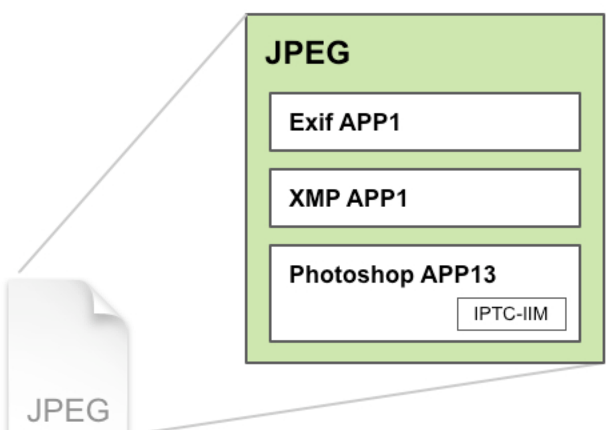
2. INTRODUCTION

Metadata, often referred to as “data about data,” provides interesting information that supplements the primary content of digital documents. Metadata has become a powerful tool to organize and search through the growing libraries of image, audio and video content that users are producing and consuming. This is especially important in the area of digital photography where, despite the increased quality and quantity of sensor elements, it is not currently practical to organize and query images based only on the millions of image pixels. Instead, it is best to use metadata properties that describe what the photo represents and where, when and how the image was taken.

Metadata is now critical in workflows ranging from consumer sharing experiences to professional-level asset management. That said, there are several complications which result from structural hierarchies required to store metadata within images:

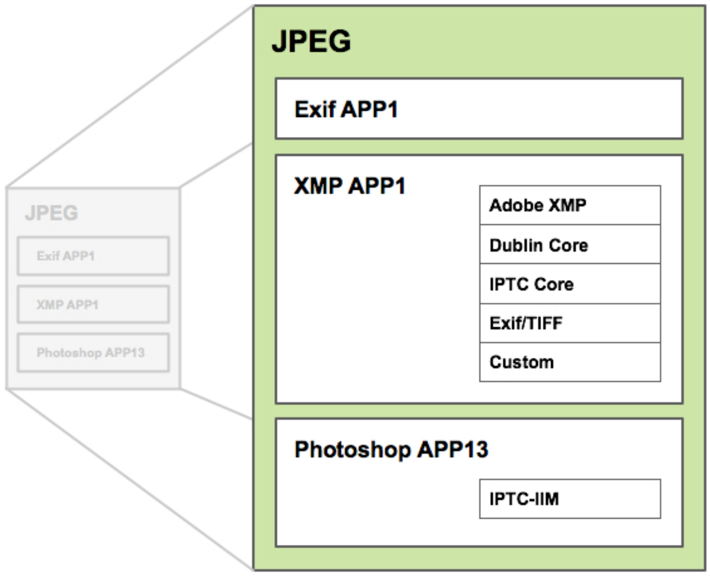
Digital images are stored in a variety of common file formats such as TIFF, JPEG and PSD as well as proprietary formats such as RAW. Each file format has distinct rules regarding how metadata formats must be stored within the file.

Within image file formats, metadata can be stored using a variety of common metadata container formats such as Exif/TIFF IFDs, Adobe XMP, Photoshop Image Resources (PSIR) and IPTC-IIM. Each metadata container format has unique rules regarding how metadata properties must be stored, ordered and encoded within the container.

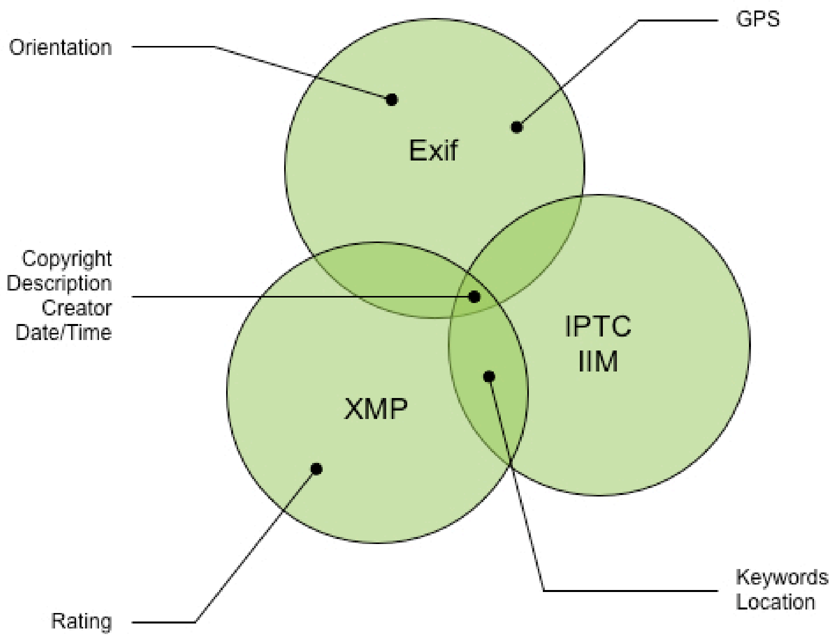


Within these metadata container formats, metadata can be stored according to various semantic groupings. Examples of these groupings are tags within Exif which relate to GPS, Dublin Core represented in XMP and IPTC-IIM’s Application Records.

Some metadata semantic groupings, such as IPTC's, are intended for use in specific user workflows and some, such as Exif's, can be stored using multiple metadata container formats.



Within metadata semantic groupings, there can be dozens of individual metadata properties. Each metadata property can require data of specific types such as strings, numbers or arrays. Some metadata properties are conventionally read-only while others can be modified by the user. Metadata properties are typically objective but some are subjective. Some useful properties, such as user ratings, have no commonly used standard storage container while others, such as copyright strings, can be stored within many containers with similar but subtly distinct semantics.



The above structural complexities have traditionally caused further complications which challenge the effective use of metadata in workflows:

- Different applications and devices have chosen to handle ambiguously or weakly-defined metadata specifications in different ways.
- Different applications and devices have chosen to follow different policies in cases where metadata can be stored in more than one standard location.
- An application or device often stores proprietary metadata, such as maker notes, within a metadata container. This practice is fragile because such private data can easily be lost when a different application modifies a file.
- Some applications and devices usurp general purpose metadata properties to address specific needs. This can cause compatibility problems for applications that correctly use those properties in accordance with the generally accepted specification.
- Some applications avoid the complexities of storing metadata within image files altogether and opt, instead, to store it in a separate file or database. This practice can easily result in the loss of metadata when a file is used across several applications.

All of these problems have led to significant frustration for users who want consistent metadata interoperability across digital imaging products and services. Manufacturers of digital imaging hardware, software and services spend substantial development resources dealing with these problems. Until practical guidance to resolve these complexities exists, these problems will continue to cost both users and industry time and resources.

2.1 About the Metadata Working Group

Based on a 2006 proposal by Microsoft, the Metadata Working Group (MWG) organization was created in 2007 by 5 founding members: Apple, Adobe, Canon, Microsoft and Nokia. Sony joined this initiative in 2008.

The goals are:

- Preservation and seamless interoperability of digital image metadata.
- Interoperability and availability of metadata to *all* applications, devices, and services.
- Enabling emerging metadata-based workflows.

The organization is based on a formal legal framework and royalty free intellectual property policy that allows member companies and other industry leaders to collaborate on a solution to the above problems. The efforts of the MWG are organized into initiatives. The continuation of MWG's first initiative (covered in this document) addresses issues of digital imaging metadata for typical consumers. Future initiatives might deal with metadata for professional photography, audio and video metadata, etc.

2.2 Scoping the work

Consumer sharing of still images has exploded with the maturing of Internet services for the storage, manipulation and sharing of pictures. However, the majority of standards related to still images are oriented towards documenting the creation of an image or towards professional usage and management of images (e.g. in print media). Little provision is made for the consumer who simply wants to share images with friends, manage their snapshots, or be creative with their photos. The intention of this document is to use and extend existing standards to address the key organizational metadata questions that most consumers have:

- **Who** is involved with this image? (who took it, who owns it, who's in it?)
- **What** is interesting about this image?
- **Where** is this image from?
- **When** was this image created or modified?

The goal of this document is to provide best practices for solving interoperability issues in the consumer space.

When we look at the "four Ws" (who, what, where, when), it is clear that this data can range from highly precise (e.g. GPS latitude/longitude) to extremely vague or context-dependent (e.g. "In my back yard"). This document does not try to solve the difficult semantic issues around this problem; rather, it tries to ensure that semantically equivalent metadata is identified across standards, and if it exists, that best practice is followed to use semantically well-defined properties for that metadata. The key notions of "reconciliation" and "rationalization" for the consumer space define the scope of this initial work.

2.3 Digital imaging metadata initiative

The scope of this effort is focused on the metadata of still images, with particular attention given to consumer workflows. While the guidance has been augmented to include some important new consumer use cases, the effort remains centered upon solving a specific set of problems, outlined below:

Issues addressed in this document

- Interoperability and preservation of metadata between processes (devices, applications, platforms and services), file formats and metadata standards.
- Overlapping fields between existing standards and schemata.
- Need for guidance on emerging consumer metadata.

These issues have been addressed by the creation of:

- A usage and data model based on common consumer use cases.
- Actor definitions of the roles each device or application plays when interacting with metadata.
- Best practices regarding how, when and where metadata should be changed in popular consumer still image file formats using existing industry metadata standards.
- Rationalization of common and important consumer metadata fields between existing standards.
- New schemata for emerging metadata that are not currently addressed in digital asset workflows.

Specific non-goals for this document

- To define new metadata structures, storage formats, or standards where appropriate alternatives already exist.

2.4 Relationship to standards organizations

There are a number of established standards, such as Exif and IPTC, which are widely used by the digital photography industry. This effort is not intended to replace existing industry standards, but rather to build on them by providing resources to improve *interoperability* and *metadata preservation* between them. This is based on significant understanding of the industry (customers, scenarios, technologies) and experience in building the products that capture, process, store, share and transmit digital photographs.

This document, “Guidelines For Handling Image Metadata”, is designed to help guide developers by providing best practices on how to create, read and modify metadata within digital images. It’s also designed to motivate the owners of metadata standards and formats to think about preservation, interoperability, and compatibility in general terms.

2.5 Definition of terms

Digest	A checksum value to help identify changes between the metadata formats
Dublin Core	A metadata element set. It includes all DCMI terms (i.e. refinements, encoding schemes and controlled vocabulary terms) intended to facilitate discovery of resources.
Exif	“Exchangeable image file format” – a standard for image file formats, jointly managed by Japan Electronics and Information Technology industries Association (JEITA) and Camera and Imaging Products Association (CIPA)
IPTC	“International Press Telecommunications Council” – creator and maintainer of metadata standards
IPTC-IIM	“Information Interchange Model” – IPTC multimedia metadata standard
IPTC Core	IPTC photo metadata standard based on XMP
IPTC Extension	Extended IPTC photo metadata standard based on XMP
JPEG	A file format, widely used in image and photography workflows
MWG	“Metadata Working Group” – Industry consortium responsible for this document
PSD	The native Adobe Photoshop file format
RDF	The “Resource Description Framework (RDF)”, described by the W3C as a “framework for representing information in the Web”, has become a general model for representing metadata
TIFF	The “Tagged Image File Format (TIFF)” is a file format to store images as well as photography.
Unicode	Unicode is an industry standard to consistently represent characters and text within modern software and hardware systems
UTF-8	UTF-8 is a byte-oriented encoding form of Unicode
URI	Universal Resource Identifier
XMP	“Extensible Metadata Platform” – multimedia metadata standard introduced by Adobe

3. USAGE AND DATA MODEL

For a better understanding of the proposed guidance in this document, this chapter introduces the notion of different *actors* that play specific *roles* in metadata processing. These definitions will be used throughout the document to discuss the rules on how to handle metadata across the different formats.

3.1 Actor definition

In the actor model, the flow of an image file is represented as a series of states between multiple applications (actors). It starts with the **Creator** actor, going through **Changer** actors and ending at the **Consumer** actor. In this model, all actors are essentially black boxes in which the processing actions, specific to each actor, are neither known nor considered important from the model's point of view. However, the state of the metadata in the image file is communicated between each phase.

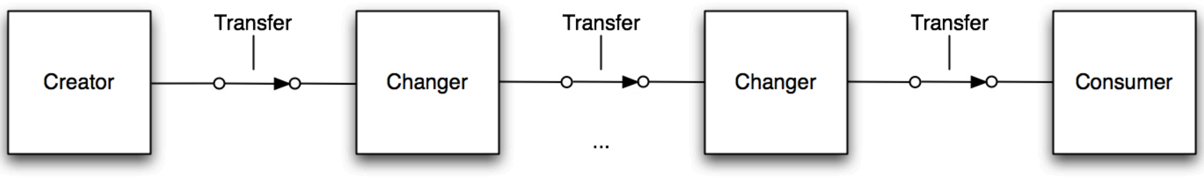


Figure 1 - Actor state diagram

An application is defined to be compliant if it reads and writes metadata in accordance with this document. There may also be non-compliant applications modifying metadata between two actors. It is not always possible to detect such modifications, so any compliant application must also accept non-compliant metadata.

This document presents the rules on how to handle a small set of selected metadata fields in a compliant manner. The roles which are defined in this document are covered for two main reasons: firstly, to attempt to clarify the purpose of the selected fields, and secondly, to show how to apply similar metadata handling to fields not covered here. An application can function in different roles at different times but every time it touches metadata, it does so only in one of these roles.

3.1.1 Creator

A **Creator** application creates the first instances of metadata in a new (image) file. It is usually (though not necessarily) the same application that creates the image data, e.g. an image processing application, a digital camera or a cell phone. One common aspect of a **Creator** application is that there is no old metadata to preserve.

A **Creator** must meet these criteria:

- It MUST have full knowledge of all metadata it is creating.
- It MUST always create standard compliant metadata in at least one form, as specified in this document.

Another scenario is an image editing application that creates a new file after consuming information from an existing one. In general, producing a new file from an existing one can be treated as a role of a **Changer** since metadata is not created but instead repurposed. Therefore please refer to the **Changer** definition below for more guidance.

3.1.2 Changer

A **Changer** application first reads metadata from an image file and then writes new or modified metadata back to a file.

The rules for an application in the **Changer** role are:

- Deletion of metadata MUST only be done with specific intent.
- It SHOULD obey rules for **Consumer** applications when reading metadata.
- It SHOULD keep all forms of metadata it modifies in sync with one another.

The first rule pertains to the preservation of metadata. Descriptive metadata, information added by a user, must only be deleted by explicit user intent. Non-descriptive metadata may however be deleted with explicit intent. It should only be modified or deleted if it is known to be inaccurate or problematic.

The second rule follows from the observation that the **Changer** application is also a **Consumer** application, so it should also observe all **Consumer** application rules.

Finally, the third rule states that whenever the **Changer** application writes new metadata fields to the file, it should keep different forms existing in the file, e.g. Exif, IPTC-IIM and XMP, in sync with one another. The first rule and third rule also apply to a changer that wishes to not write all forms of a metadata item. Writing one form and deleting other forms is a legitimate intentional deletion, done to avoid unsynchronized forms in the file.

3.1.3 Consumer

A **Consumer** application only reads metadata from the image file. It may use metadata for display purposes, searching, content organization, etc. but it never modifies the metadata in the file itself.

General rules for **Consumer** application are:

- It MUST reconcile between different forms of metadata in the image.
- It MUST use metadata according to the semantics defined for each field.

The first rule says that a **Consumer** application must process metadata according to policies defined in this document and then only the reconciled data is used further before it is presented to the end-user. This may involve, for example, resolving possibly conflicting information between Exif, IPTC-IIM and XMP versions of a metadata field existing in the image file.

The second rule says that applications must understand the semantics of desired metadata fields and use them appropriately. For example, in order to reconcile different forms of metadata, the application must know the semantics of the metadata in question.

In summary, the **Consumer** application treats image files as read-only so the state of the metadata remains unchanged.

Tools designed to display technical details about the format and content of the file's metadata, but not intended to primarily express metadata semantic meaning, are not required to be compliant **Consumer** applications.

4. METADATA MANAGEMENT

Metadata is an essential part of image and photography based workflows. Cameras capture device metadata while taking pictures. Operating systems and other software subsequently read metadata to build up catalogs and offer effective search capabilities. In addition to this, the user is able to enhance this workflow with their own metadata that may be stored either inside the file or within caching or database systems.

In the context of consumer image-based workflows, the existence of different metadata standards leads to interoperability issues when using various devices, operating systems and software tools. Although the majority of metadata properties are unique, there are a number of properties which overlap across several metadata standards and cause interoperability issues as a consequence.

The goal of this section is to identify those overlapping properties and provide guidance on how to handle them correctly across the different metadata formats.

After a brief overview of existing metadata standards, this chapter introduces the most common metadata properties in the context of consumer workflows. To ensure best interoperability across software and hardware systems, a general reconciliation mechanism is then discussed. Finally, the chapter will close with a detailed analysis of each focus area and discuss specific technical issues and obstacles.

4.1 Existing metadata standards

This section gives an overview of the existing metadata formats. As described in the introduction, this document will focus on photography workflows in the context of the consumer, so the choice of discussed metadata formats covers Exif, IPTC-IIM and XMP.

Exif - Exchangeable image file format

The Exif standard has been jointly managed by the Japan Electronics and Information Technology industries Association (JEITA[1]) and Camera and Imaging Products Association (CIPA). In particular, the Exif image interchange format defines a set of TIFF tags that describe photographic images, and is widely used by digital cameras. Exif metadata can be found in TIFF, JPEG, and PSD files.

DCF - Design rule for Camera File system

As digital still cameras (DSC) have grown enormously in popularity, there is a growing need for the direct exchange of images between cameras and other equipment, allowing pictures taken on one camera to be viewed on another, or to be sent to a printer. The DCF specification is intended to simplify the interchange of files between digital still cameras and other equipment. To this end, it specifies rules for recording, reading and handling image files and other related files used by DSC and other equipment. Among other things, DCF defines a subset of Exif where some properties are optional in Exif but required in DCF.

⇨ http://www.jeita.or.jp

IPTC – International Press Telecommunication Council

IPTC, based in London, UK, is a consortium of the world's major news agencies, news publishers and news industry vendors. It develops and maintains technical standards for improved news exchange that are used by virtually every major news organization in the world.

In 1979, the first IPTC standard was text-only and defined to protect the interest of the telecommunications industry.

⇨ http://www.iptc.org

Later, in 1991, a new standard, the **"Information Interchange Model" (IIM)**, was created. IIM is an envelope format for transmitting news text documents and photos, and it defines the so-called "IPTC headers" which now exist in many photo files, having been inserted by Adobe Photoshop and similar software.

⇨ http://www.iptc.org/IIM

After Adobe had introduced XMP in 2001, the IPTC Core and IPTC Extension standards have adopted XMP as the successor to the IIM-based "IPTC header" used to describe millions of professional digital images.

⇨ http://www.iptc.org/photometadata

XMP - Adobe's Extensible Metadata Platform

XMP is a labeling technology that allows you to embed metadata into a file itself. With XMP, desktop applications and back-end publishing systems gain a common method for capturing, sharing, and leveraging this valuable metadata - opening the door for more efficient job processing, workflow automation, rights management and many other potential improvements. XMP standardizes the definition, creation and processing of extensible metadata.

XMP defines a metadata model that can be used with any defined set of metadata items. XMP also defines particular schemata for basic properties useful for recording the history of a resource as it is processed; from being photographed, scanned, or authored as text, through photo editing steps (such as cropping or color adjustment), to assembly into a final image. XMP allows each software program or device along the way to add its own information to a digital resource, which can then be retained in the final digital file.

XMP is serialized in XML and stored using a subset of the W3C Resource Description Framework (RDF). Therefore, customers can easily define their own custom properties and namespaces to embed arbitrary information into the file.

⇨ http://www.adobe.com/products/xmp

4.2 Metadata formats within image files

There are three metadata formats widely used in the industry:

- Exif
- IPTC-IIM
- XMP

Within this section, guidance for reconciling and writing metadata across these metadata formats is discussed. Refer to the respective specification for details on reading, writing, or modifying metadata in each format.

4.2.1 Handling a single metadata format

In the simplest scenario, a given metadata property is only defined in a single metadata format. This is, for example, true for the rating property - this value should always be read and written into the corresponding XMP (xmp:Rating) field. No further reconciliation is necessary. Also, there are a variety of properties defined in Exif (device properties) or in IPTC-IIM (workflow properties) that are unique to the container and won't be reconciled amongst the other formats.

4.2.2 Handling multiple metadata formats

Dealing with more than one metadata format makes it challenging to determine the correct behavior for handling the particular property values. The main difficulty is the evolution of metadata representations and standards where older applications are not aware of newer practices. This can happen within a standard, such as the introduction of Unicode storage for IPTC-IIM. It can also happen across standards, such as with the introduction of XMP. Inconsistent implementations across software tools, encoding requirements, as well as size limitations on metadata properties cause additional challenges.

The properties described earlier have been identified as the most relevant in the consumer workflows today. However, they also serve another purpose in this document. Nearly all of them are defined in more than one metadata format, so they are good candidates for helping to understand the reconciliation issues between the various formats. In other words, if the problems for these properties are well understood, all other metadata properties can be handled accordingly.

Here is a simplified view of metadata for which guidance is being provided:

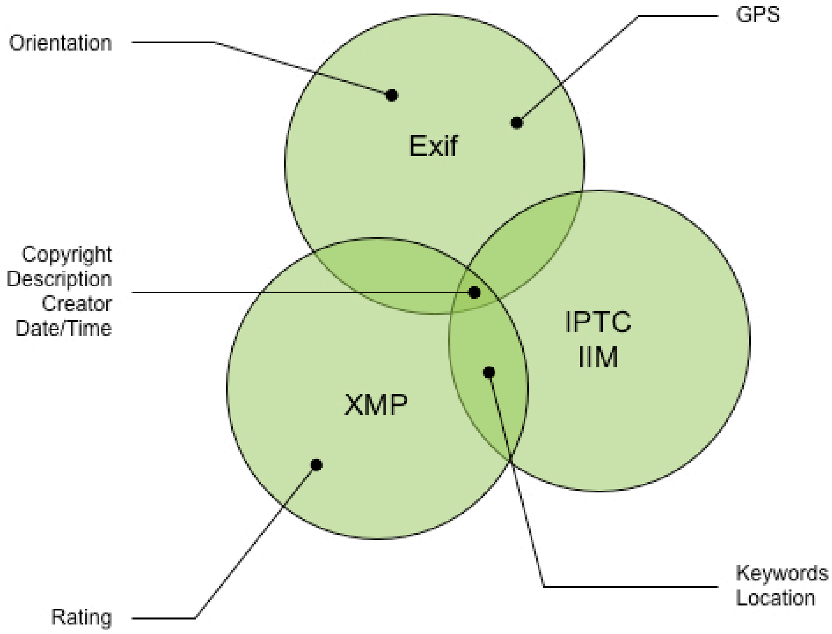


Figure 2 - Metadata defined in more than one format

It is evident that there are only a few properties defined in more than one metadata format. Actually only four are available in Exif, IPTC-IIM **and** XMP (Copyright, Description, Creator and Date/Time).

To ensure interoperability between existing and upcoming hardware and software solutions, the following sections give you an overview of how to handle the different metadata properties in the context of the actor/role definitions.

But before going there, the next chapter will provide some more background information on the specific relationship between the metadata formats to better understand the overall picture.

4.2.2.1 Exif and IPTC-IIM in the context of XMP

The following diagram presents a different perspective on metadata usage:

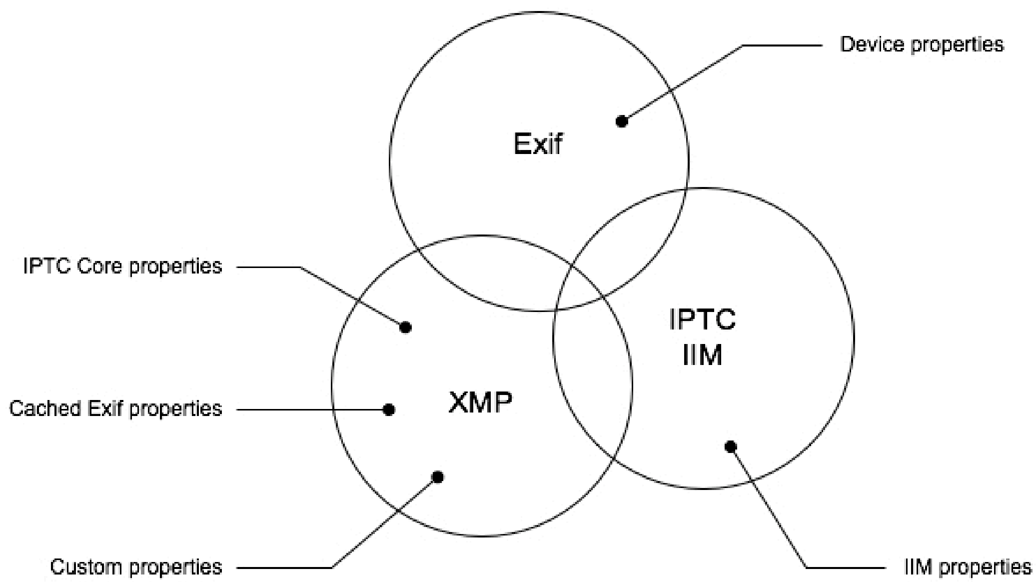


Figure 3 - Usage scenarios of Exif, IIM and XMP

Beside the fact that some native Exif and IPTC-IIM properties are mapped to corresponding XMP properties, some popular applications that pre-date this guidance also replicate a large number of other Exif properties into the XMP. To better understand the different use cases, the following two chapters will put these properties into the context of this document.

Exif within XMP

The most recent (as of mid-2010) XMP specification describes the usage of Exif/TIFF properties within XMP itself. Both Exif (http://ns.adobe.com/exif/1.0/) and TIFF (http://ns.adobe.com/tiff/1.0/) namespaces have been defined so that corresponding Exif properties can be stored. This is particularly useful if Exif properties need to be stored but the file format does not support native Exif (e.g. PNG).

In the case of file formats that **do** support Exif however, the current XMP specification describes mechanisms to reconcile data between the native Exif values and the mapped Exif properties in XMP (see “TIFF and Exif digests” under section “Reconciling metadata properties” in the XMP specification).

However, this document changes this earlier XMP guidance and recommends that Exif and Tiff device properties only be mapped into XMP in the case the file format does not support Exif natively. For more details, please see section “Handling Exif and XMP” below.

IPTC within XMP

In contrast to the earlier IPTC-IIM specification, the most recent IPTC Core specification allows storing IPTC properties as XMP. Most of the properties are mapped to existing standard namespaces but for those where this was not possible, a new namespace “http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/” has been introduced. The IPTC group encourages people to move from IPTC-IIM to its newer IPTC Core / IPTC Extension standard.

With that said, this document focuses on the interoperability between existing applications and is mainly concerned with the reconciliation between the earlier IPTC-IIM standard and XMP, concentrating on the areas discussed in this document. There is no “reconciliation between XMP and IPTC Core” - everything is in XMP.

4.2.3 Metadata reconciliation guidance

The process of handling metadata values from the various metadata formats is basically divided into three different scenarios which will be discussed in the following chapters:

- Read/Write Exif and XMP metadata
- Read/Write IPTC-IIM and XMP metadata
- Read/Write Exif, IPTC-IIM and XMP metadata

4.2.3.1 Handling Exif and XMP

This chapter discusses reconciliation guidance between Exif and XMP:

Reading Exif and XMP

Only a few properties are actually mapped between Exif and XMP and are therefore relevant for reconciliation. These are:

- Date/Time
- Description
- Copyright
- Creator

Since only a few properties are mapped between Exif and XMP, they are dealt with on a property-by-property basis. Unlike IPTC-IIM, as seen later, there is no advantage in using a checksum value to detect changes to the Exif. Especially for consumer use, there is generally no loss of fidelity when preferring Exif over XMP.

Here is a detailed look:

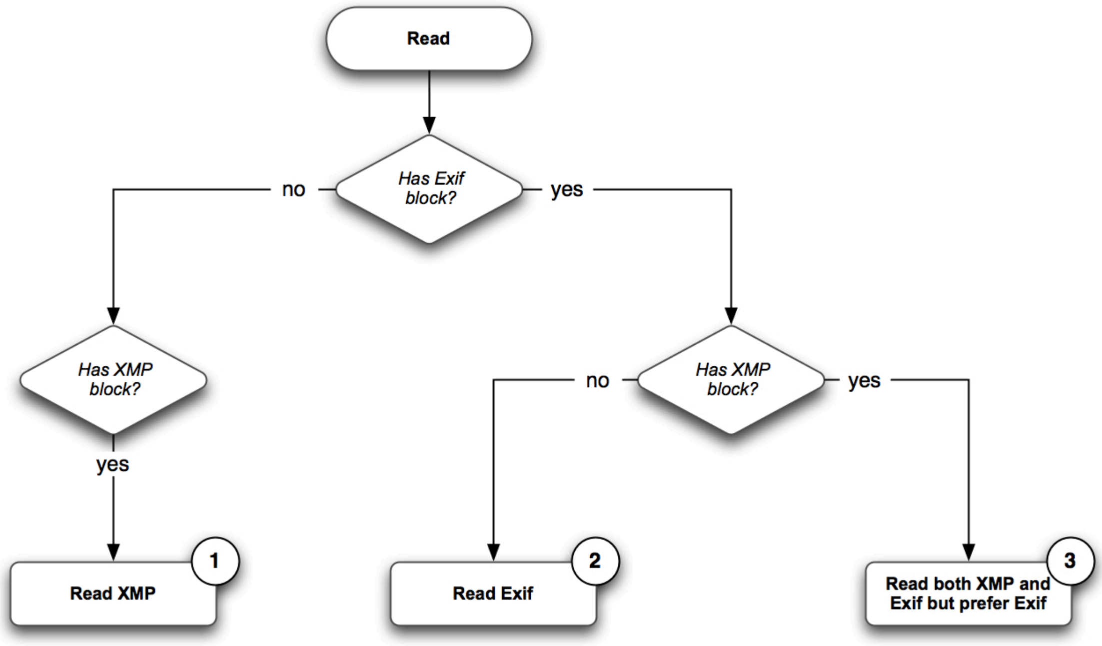


Figure 4 - Read guidance Exif/TIFF

If either Exif **or** XMP is available (1 & 2) reading metadata is straightforward.

Note: Today, there are two scenarios where Exif metadata is being mapped into XMP:

- Exif native properties mapped to respective XMP properties (e.g. Exif Copyright ⇔ XMP (dc:rights))
- Exif and TIFF device properties duplicated into specific “exif:” or “tiff:” namespaces (e.g. Exif ApertureValue ⇔ XMP (exif:ApertureValue))

In particular, for scenario (1), this means Exif and Tiff device properties SHOULD be read directly from the respective “exif:” and “tiff:” XMP namespaces. This is the case when the file does not natively support Exif.

The scenario “Read both XMP and Exif but prefer Exif” (3) is the most interesting one: In general, a **Consumer** SHOULD prefer the values from Exif. As we'll see in more detail below, Exif can be preferred when reading because it does not have encoding or length limitations with respect to XMP. The policy for **Changers** ensures that newer applications write consistent values; preferring Exif when reading supports older applications.

It is, however, important to carefully read and follow the individual property mappings described in section “5. Metadata Guidelines” of this document. For example, the XMP (dc:description) value supports multiple languages, the corresponding Exif maps to a specific one of these.

The following diagram explains how to read Exif and XMP on a per property level in more detail:

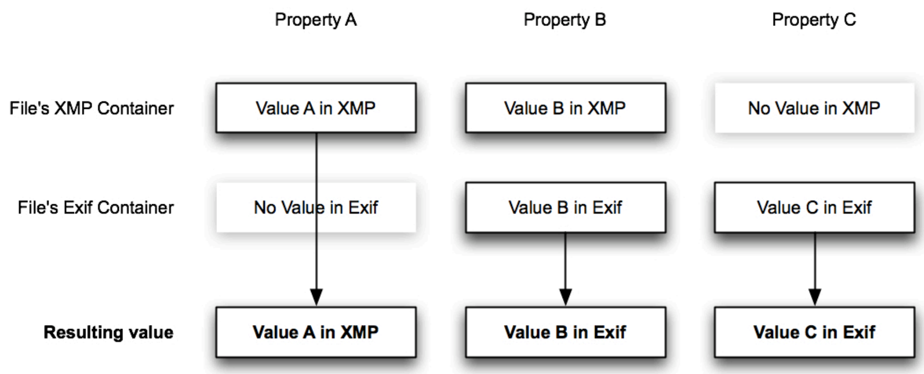


Figure 5 - Reconciling properties for Exif and XMP

For broader compatibility with non-compliant **Creators** and **Changers**, a **Consumer** SHOULD verify whether Exif text values are valid UTF-8. If not, a **Consumer** MAY assume the value is in a "local encoding" and convert it to UTF-8 as described under "Text encodings in read and write scenarios" below.

Writing Exif and XMP

In the context of the actor definitions, the following rules describe the guidance on how to write XMP and/or Exif:

Creator

- XMP metadata MAY be created if a property can be written in both locations otherwise it MUST be created (which is true for file formats where Exif is not defined).
- If no XMP is written, Exif metadata MUST be created.

Changer

- Exif and XMP metadata SHOULD be consumed according the reconciliation guidance described earlier (see "Reading Exif and XMP" above).
- When the file format supports both Exif and XMP, a **Changer** SHOULD update both forms of a value. If only one form is updated, an existing value in the other form MUST be removed.
- In the case the file format does support Exif natively, Exif and TIFF device properties (e.g. XResolution, YResolution, WhitePoint, etc.) SHOULD NOT be duplicated in the XMP exif: and tiff: namespaces.
- Exif metadata is formatted as a TIFF stream, even in JPEG files. TIFF streams have an explicit indication of being big endian or little endian. A **Changer** SHOULD preserve the existing byte-order.

- Exif string values SHOULD be written as UTF-8. However, clients MAY write ASCII to allow broader interoperability[2].
- A checksum value for Exif/TIFF SHOULD NOT be written into the XMP.
- If no existing data is in the file, the **Creator** guidance SHOULD be followed.

4.2.3.2 Handling IPTC-IIM and XMP

This chapter discusses reconciliation guidance between IPTC-IIM and XMP:

Reading IPTC-IIM and XMP

The use of IPTC-IIM is significant in professional workflows, and is also present in some consumer oriented tools. Although this document only directly addresses a few IPTC-IIM fields, there are several dozen in professional use. The IPTC-IIM values have length limitations and often character encoding issues that can make a conversion from XMP to IPTC-IIM be lossy.

For efficiency, and to avoid certain character encoding problems, a checksum (or digest) is used to detect overall changes to the IPTC-IIM values by non-compliant **Changers** - specifically those unaware of XMP. This checksum detects that something has changed in the IPTC-IIM block as a whole, but not specifically what has changed. Further checks are then required to detect individual property changes.

The checksum value is an MD5 hash of the entire IPTC-IIM block, and is stored as a 16 byte binary value in Photoshop Image Resource (PSIR) 1061 (see "Writing IPTC-IIM and XMP" for more details). The checksum MUST be computed and stored when a **Creator** or **Changer** writes XMP and IPTC-IIM simultaneously. A **Consumer** MUST use the checksum as described below when reading XMP and IPTC-IIM.

[2] *It is understood that writing UTF-8 in Exif is formally in violation of the Exif specification, which requires 7-bit ASCII in most cases. Some devices (cameras and printers) will not be able to display non-ASCII characters.*

Now let's have a look at the **Consumer** guidance on how to read IPTC-IIM related metadata first:

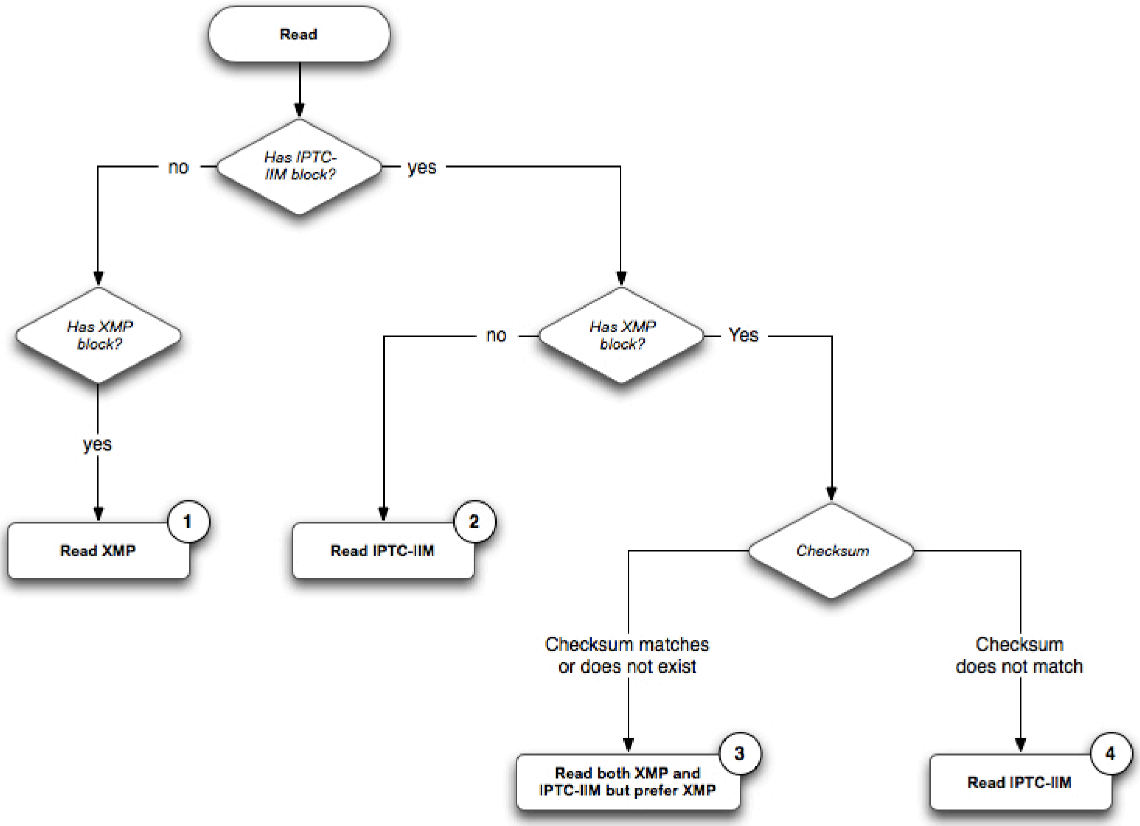


Figure 6 - Read guidance IPTC-IIM

In the case when either IPTC-IIM or XMP is available the read scenario is trivial (1 & 2).

However, scenarios (3) and (4) are more complex and require further explanation:

In scenario (3) either the checksum exists but matches the IPTC-IIM block **or** the checksum does not exist. In either case the following rules apply:

- Any existing XMP value is assumed to be more relevant and SHOULD be preferred.
- If an XMP value is missing then the IPTC-IIM value SHOULD be used.

The following diagram explains how to read IPTC-IIM and XMP on a per property level in this scenario:

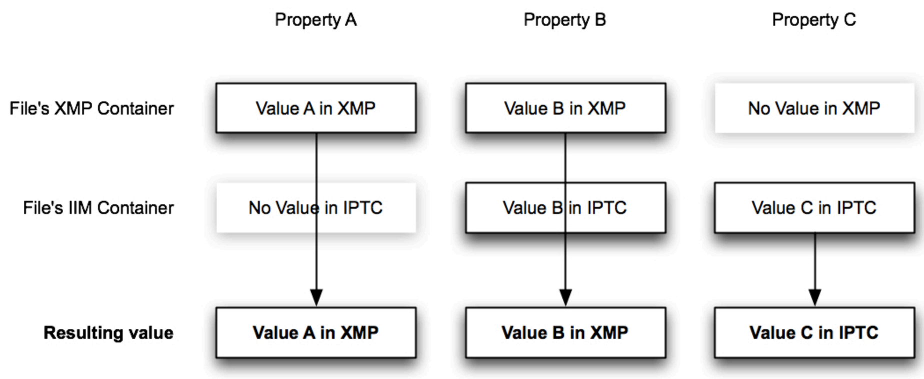


Figure 7 – Reconciling properties for IPTC-IIM and XMP

Finally, scenario (4) occurs when a non-compliant (XMP-unaware) **Changer** has exclusively modified the IPTC-IIM block. In this case a **Consumer** SHOULD check each property to decide if the IPTC-IIM or XMP value is used. This approach prevents loss of information in unchanged values due to truncation or character encoding. To do the check, the XMP value is used to create a predicted IPTC-IIM value, taking value truncation and character encoding into account. If the predicted and actual IPTC-IIM values match, then the XMP value is used. Otherwise the IPTC-IIM value is used.

A **Consumer** MUST honor the encoding information provided by any IPTC-IIM Coded Character Set 1:90 DataSet that specifies UTF-8 - it SHOULD honor other 1:90 encodings. If this is not present, a **Consumer** MAY assume the value is in a “local encoding” and convert it to UTF-8 as described under “Text encodings in read and write scenarios”.

Writing IPTC-IIM and XMP

The following rules describe the guidance on how to write XMP and/or IPTC-IIM:

Creator

- SHOULD NOT create IPTC-IIM unless backward compatibility with non-compliant **Consumers** that don't read XMP is a requirement – otherwise SHOULD write XMP.
- If IPTC-IIM and XMP are both written, a **Creator** MUST create the checksum value as described earlier.

Changer

- XMP and IPTC-IIM SHOULD be consumed according the reconciliation guidance described above.
- If IPTC-IIM is already in the file, a **Changer** SHOULD write data back to the file in both XMP and IPTC-IIM – otherwise only XMP SHOULD be written.

- IPTC-IIM SHOULD be written using the Coded Character Set (1:90) as UTF-8 (see "Section 1.6 Coded Character Set" in the IIM specification).
- If the IPTC-IIM has not been written in UTF-8 before, a robust **Changer** SHOULD convert all properties to UTF-8 and write the corresponding identifier for UTF-8 to the 1:90 DataSet.
- If IPTC-IIM and XMP are both present, whether changed or not, a **Changer** MUST create or update the checksum value as described earlier.
- If no existing metadata is in the file, the **Creator** guidance SHOULD be followed.

IPTC-IIM checksum

In the example of IPTC-IIM, the checksum MUST be calculated over the entire IIM block after values have been converted from XMP. The checksum itself MUST be stored in the Photoshop Image Resource (PSIR) 1061 resource as a 16-byte binary value representing the MD5 hash over the whole IIM block.

Example:

```
PSIR (1061) = 0ED63323337C50BF1E3BA76F6BB2122F
```

4.2.3.3 Handling Exif/TIFF, IPTC-IIM and XMP metadata

This chapter discusses reconciliation guidance between Exif, IPTC-IIM and XMP:

There are four properties that are defined in all metadata formats being discussed. Because the reconciliation guidance is specific to each property, please see section "Metadata Guidelines" later in this document for more details. If there is a conflict between Exif and IPTC-IIM, a **Consumer** SHOULD prefer Exif in the case the IPTC-IIM checksum matches or does not exist and SHOULD prefer IPTC-IIM in the case the checksum does not match. A string property that is comprised of only spaces or only nul characters MUST be treated as non-existent.

Upon writing Exif metadata, a **Changer** MUST update all formats that were originally present in the file. If not all of the formats were originally present, a **Changer** MAY choose to write the complete set.

4.2.3.4 More complex reconciliation in popular image formats

Finding and interpreting the metadata embedded in JPEG, TIFF, and PSD files is complicated by the fact that all three file formats contain the same kinds of metadata (XMP, Exif/TIFF, and IPTC-IIM), but store it slightly differently.

For example, all of the kinds of metadata can be contained in Photoshop Image Resources (PSIRs), and all three file formats (JPEG, TIFF, and PSD) can contain PSIRs. However, the specific contents of the PSIRs are different when contained in different image file formats. Each type of metadata is stored inside the PSIR for some file formats, and separately for others.

However, the recursive embedding of metadata formats is more a theoretical possibility, so this document will simplify this process by identifying the three most relevant places to find Exif, IPTC-IIM and XMP (highlighted below).

Here are illustrations of the various image file formats:

JPEG file format

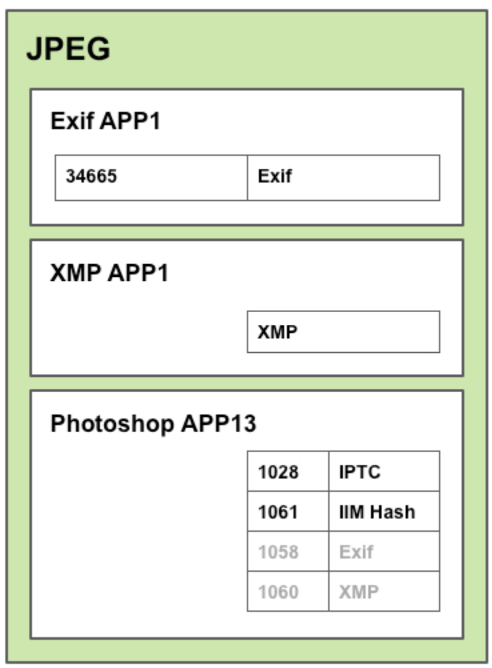


Figure 8 - JPEG file format

XMP SHOULD be read from the XMP APP1 section, IPTC-IIM SHOULD be read from the image resource block in Photoshop APP13 (1028) and finally Exif SHOULD be read from the Exif APP1 section. All other occurrences SHOULD be ignored. Please note that the APP markers SHOULD be written according to the Exif specification. In particular, Exif APP1 MUST follow immediately after SOI. If this is not the case, current camera models may not show Exif metadata correctly.

TIFF file format

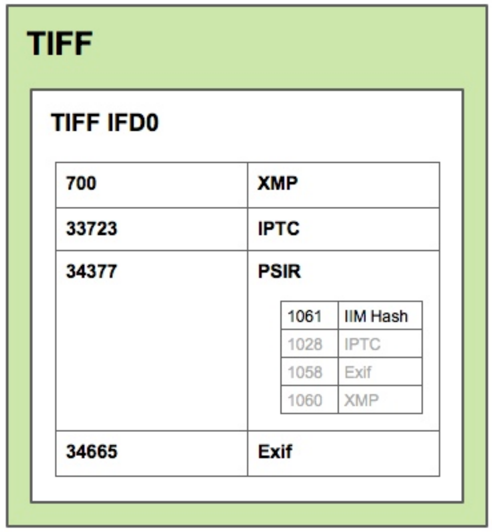


Figure 9 - TIFF file format

The *TIFF IFD0* contains “Exif” (34665), “IPTC” (33723) and “XMP” (700) and SHOULD be used. The IPTC-IIM checksum is stored within the “PSIR” block (34377).

PSD file format

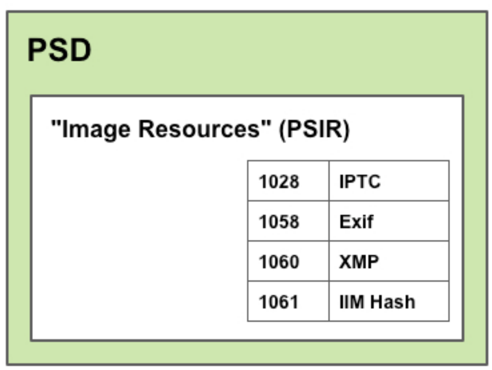


Figure 10 - PSD file format

The respective PSIRs - “Exif” (1058), “IPTC” (1028) and “XMP” (1060) SHOULD be accessed directly to read and write the various metadata formats.

Obviously, there are other file formats used by consumers including GIF, PNG, DNG, JPEG XR, etc. These files will not be discussed in this document.

Text encodings in read and write scenarios

It is important to understand text encoding issues when reading and writing string metadata properties. The encoding defines the mapping between numerical byte values and user-readable glyphs. It also defines the limits on what glyphs of which languages a byte sequence can represent. It is critical to know the encoding of a string property in order to correctly display the string to the user. If a string is displayed with the wrong encoding it will likely appear as a nonsensical string of glyphs.

The text encoding guidelines for **writing metadata** are simple and provide Unicode output in all cases.

- XMP metadata MUST be written as Unicode in the form appropriate to the file. For JPEG, TIFF and PSD files this is UTF-8.
- IPTC-IIM metadata MUST be written as UTF-8, and MUST include a 1:90 DataSet indicating the use of UTF-8.
- Exif tags documented in the Exif specification as type ASCII SHOULD be written as UTF-8. Note that 7-bit ASCII is a proper subset of UTF-8. They MAY be written as 7-bit ASCII, with appropriate trimming for out of range bytes. These tags MUST NOT be written in some other encoding.
- Exif tags such as UserComment that have an encoding prefix SHOULD be written as ASCII if all bytes fit in the 7-bit range 0..127, and MUST be written in some other encoding if any byte is outside the 7-bit range. When not using ASCII, Unicode UTF-16 SHOULD be written. The UTF-16 byte order nature SHOULD match the containing TIFF stream, big endian if the TIFF begins with “MM” and little endian if the TIFF begins with “II”.

Of course, a device or application could have a mode of operation that writes text in other encodings for compatibility with older workflows. Such older workflows are, by definition, not MWG-compliant however.

The text encoding guidelines for **reading metadata** recognize the desired Unicode and also accommodate other cases for IIM and Exif that might be written by non-compliant devices or applications.

- XMP metadata MUST be read as Unicode in the form appropriate to the file. For JPEG, TIFF and PSD files this is UTF-8.
- IPTC-IIM metadata MUST be read as UTF-8 if a 1:90 DataSet is present indicating the use of UTF-8.
- IIM metadata MUST be read using the appropriate encoding if a 1:90 DataSet is present indicating the use of an encoding other than UTF-8, otherwise it MAY be ignored.
- IIM metadata SHOULD be read as described below when no 1:90 DataSet is present.
- Exif tags of type ASCII SHOULD be read as described below.
- Exif tags such as UserComment that have an encoding prefix MUST be read as UTF-16 if the encoding is Unicode, and the UTF-16 byte order nature SHOULD be determined as described below. These tags SHOULD be read as described below if the encoding is ASCII. These tags MUST be read as JIS X208-1990 if the encoding is

JIS, otherwise they MAY be ignored. These tags SHOULD be ignored if the encoding is Undefined.

All Unicode characters MUST be allowed when reading or writing UTF-16. The use of 16-bit Unicode MUST NOT be restricted to the UCS-2 subset.

A leading U+FEFF byte order mark (BOM) SHOULD be written as part of a UTF-16 value for Exif tags such as UserComment. A U+FEFF BOM SHOULD NOT be written for individual UTF-8 metadata items. A U+FEFF BOM SHOULD be ignored if encountered when reading UTF-16 or UTF-8, except for use in determining UTF-16 byte order as described below. A U+FEFF BOM SHOULD be removed from the value for encoding conversion, display, and editing purposes.

The following heuristics SHOULD be applied when an 8-bit character encoding is not known. This applies to the case of IIM without a 1:90 DataSet, to Exif tags of type ASCII, and to Exif tags such as UserComment when the encoding is reported as ASCII.

- If all bytes are in the range 0..127 then the encoding is ASCII.
- If the entire sequence is valid UTF-8 then the encoding is UTF-8.
- Otherwise assume some reasonable fallback encoding, or ignore the value.

The choice of a reasonable fallback encoding is application and workflow dependent. It can be determined by querying the locale information of the host device or the user's preference.

It is also worth mentioning that a byte sequence appearing to be valid UTF-8 is not 100% guaranteed to actually be UTF-8. There are possible edge cases where other encodings might produce such a byte sequence. Nevertheless, the UTF-8 test is highly reliable and allowing use of UTF-8 as described above is very beneficial to users.

The following heuristics SHOULD be applied when the big or little endian nature of UTF-16 text needs to be determined. These apply to a single item at a time, not uniformly to all UTF-16 text.

- If a leading U+FEFF BOM is present, that indicates the byte order.
- If only one of the byte orders is valid UTF-16, the valid form is the byte order. This MUST take into account surrogate pairs, and it MAY take into account specific invalid Unicode characters.
- Count the number of unique values in the first and second bytes of the 16-bit storage units. The correct byte order is the one with the fewer unique values in the high order part.
- Otherwise use the overall TIFF stream byte order.

Time-zone handling

The handling of date/time values, and especially time zones, is conceptually easy but requires some care to avoid confusing users. The potential problems typically stem from the differing representations of date/time values in Exif and XMP. (For our purposes here the Exif sub-seconds portions are ignored, but they are, of course, taken into account in software conversions.)

Exif date/time values such as DateTimeOriginal do not contain time zone information. The camera is presumably in an appropriate local time when a photograph is taken, but there is no indication in the Exif metadata of what that time zone was. The photograph's time zone MUST NOT be presumed to be the same as that of a computer later used to process the photograph.

The XMP specification formats date/time values according to the Date and Time (W3C) document. In this standard, a time zone designator is required if any time information is present. A date-only value is allowed. The XMP specification has been recently revised to make the time zone designator be optional.

The representation of time zone as an offset from UTC can be ambiguous with regard to daylight savings time (DST). While date information can provide a strong hint, the use of DST is not universal and the date checking is complicated by changing rules for the start and end of DST in various locations. While these issues are beyond the scope of this document, they may be addressed in a future revision.

The following general behaviors are recommended for time zone handling:

- A **Consumer** MUST NOT arbitrarily add a time zone. E.g. when importing Exif DateTimeOriginal to XMP (xmp:CreateDate), use a zone-less form for the corresponding XMP value.
- A **Changer** MUST NOT implicitly add a time zone when editing values. It is okay to be explicit about time zones if desired. Consider the typical case of correcting DateTimeOriginal values for an incorrectly set camera time. This must not be implicitly done as though the new time were in the computer's time zone.
- If the Exif contains the GPSDateStamp and GPSTimeStamp tags, software MAY use that information to infer a time zone. This should be done with care, e.g. verifying that the DateTimeOriginal plus inferred offset is within a few seconds of the GPS date and time.
- When time zone information is available, XMP values SHOULD be stored using the local+offset form rather than the “Zulu” form (for example, use “2008-04-30T12:34:56-06:00” instead of “2008-04-30T18:34:56Z”). The local+offset form carries additional information and the Zulu value can easily be determined as required, e.g. for sorting in a UI.
- A user interface MAY display time zone information if available; however, related functionality MUST NOT convert a time to the computer's local time for display.
- According to the Exif specification, missing information SHOULD be filled up with spaces in the Exif values.

In summary, time-zone information MUST NOT be implicitly added and existing values should be preserved.

5. METADATA GUIDELINES

This section will discuss some of the most relevant metadata areas in the consumer workflow today. The selection will mainly serve the purpose of discussing the most important metadata fields, but due to the fact that information in these areas can be found in multiple metadata sources, it will also act as a model for other properties as defined in Exif, IPTC-IIM and XMP. Each section will provide detailed guidance on how to handle related metadata containers.

5.1 Keywords

Keywords are widely used across software applications today and are also called “tags” by some applications and services. Since so many existing applications allow for keyword display and editing it is now often misused. Keyword properties are no longer strictly for keywords; applications overload them with general-purpose information for purposes such as workflow or task management. Recent cameras also have the ability to assign tags automatically while shooting pictures. Keywords tend to be user customizable, although in the case of devices they are sometimes fixed.

Representation

Information for the keyword property is available in:

IPTC Keywords (IIM 2:25, 0x0219) and

XMP (dc:subject[*])

Guidance

IPTC Keywords is mapped to XMP (dc:subject).

IPTC Keywords MAY be repeated, each mapping to one of the elements in the XMP (dc:subject) array. Keyword lists SHOULD be completely replaced while reconciling.

For more information see chapter “Handling IPTC-IIM and XMP” earlier in this document.

Restrictions

Each IPTC-IIM keyword is limited to 64 bytes.

Notes

Keyword properties usually do not retain the semantics of the keyword value itself. E.g. the information that “San Francisco” is a location will be lost. XMP provides the ability to add qualifiers for each keyword to define such a semantic. For future extensibility, these attributes SHOULD be preserved on any keyword manipulation.

For more details such as type information and syntax, see the respective specifications.

5.2 Description

This area defines the textual description of a resource's content. Also known as “user comment”, “caption”, “abstract” or “description”. Today, this information is represented in different ways; sometimes integrated and displayed as one field – at other times revealed separately. This document combines the different sources into one overall representation, called “Description”.

Representation

Information for the description property is available in the following properties:

Exif ImageDescription (270, 0x010E)

IPTC Caption (IIM 2:120, 0x0278)

XMP (dc:description[“x-default”])

Guidance

Exif ImageDescription, IPTC Caption, and XMP (dc:description) are mapped together.

For more information see chapter “Handling Exif/TIFF, IPTC-IIM and XMP metadata” earlier in this document.

Restrictions

Length limitation in IPTC-IIM is 2000 bytes.

Notes

In XMP, the description can be represented in multiple languages. In this document only the “x-default” value will be discussed and used. Clients MAY support the full range of localized values.

For more details such as type information and syntax, see the respective specifications.

5.3 Date/Time

There's a lot of confusion about date/time handling. In addition to a variety of date/time values stored within a file's metadata, creation and modification values are also stored by the file system - both the computer's file system and that of a camera's media card.

In general, date/time metadata is being used to describe the following scenarios:

- Date/time *original* specifies when a photo was taken
- Date/time *digitized* specifies when an image was digitized
- Date/time *modified* specifies when a file was modified by the user

Date/time *original* and date/time *digitized* are usually added by the devices (cameras, scanners, etc.) but in other scenarios the user needs to define these values manually. This can happen, for example, if an old photograph is scanned-in (digitized) and the user wishes to specify in the metadata the date the original photo was taken. The date/time *modified* value will be changed by software and operating systems on subsequent edits of the file.

This document focuses on the date/time *original* value, since that is generally the most important aspect for the consumer.

Representation

Information for Date/Time (Original) is available in the following properties:

Original Date/Time – Creation date of the intellectual content (e.g. the photograph), rather than the creation date of the content being shown

Exif DateTimeOriginal (36867, 0x9003) and SubSecTimeOriginal (37521, 0x9291)

IPTC DateCreated (IIM 2:55, 0x0237) and TimeCreated (IIM 2:60, 0x023C)

XMP (photoshop:DateCreated)

Digitized Date/Time – Creation date of the digital representation

Exif DateTimeDigitized (36868, 0x9004) and SubSecTimeDigitized (37522, 0x9292)

IPTC DigitalCreationDate (IIM 2:62, 0x023E) and DigitalCreationTime (IIM 2:63, 0x023F)

XMP (xmp:CreateDate)

Modification Date/Time – Modification date of the digital image file

Exif DateTime (306, 0x132) and SubSecTime (37520, 0x9290)

XMP (xmp:ModifyDate)

Guidance

The general mapping is described above. For more information, see chapter “Handling Exif/TIFF, IPTC-IIM and XMP metadata” earlier in this document.

Restrictions

Exif DateTime does not contain time-zone information.

Notes

Changes to XMP (xmp:CreateDate), for example to fix an incorrect camera setting, SHOULD be exported to Exif. If both XMP (xmp:CreateDate) and Exif DateTimeOriginal are missing, but Exif DateTimeDigitized (36868, 0x9004) exists, Exif DateTimeDigitized SHOULD be used to create an initial XMP (xmp:CreateDate). This is also true in the case that only IPTC DateCreated is available.

Exif DateTime is mapped to XMP (xmp:ModifyDate). Any change to the file SHOULD cause both to be updated.

The above guidance implies that Exif sub-second and IPTC-IIM time properties are being handled according to the corresponding main properties. DCF specification requires DateTimeOriginal and DateTimeDigitized; the Exif specification recommends DateTime.

For more details such as type information and syntax, see the respective specifications.

5.4 Orientation

A major sticking point in image-based workflows is the correct handling of orientation. Today, various software tools do not follow a consistent set of rules in interpreting and changing the related metadata field in conjunction with the primary and/or thumbnail images - this leads to an incorrectly rotated display of the image. There are three scenarios of interest:

- Capturing orientation information on the devices
- Changing the orientation of an image by using an asset management tool
- Editing the image and rotating the pixels

Representation

The Orientation is represented in Exif Orientation (274, 0x0112).

Guidance

An image **Creator** MUST include an orientation tag in the Primary Image if the image raster data is intended to be displayed in any orientation other than the Normal (value 1) case where the 0th row represents the visual top of the image, and the 0th column represents the visual left-hand side.

An image **Creator** MAY include an optional Thumbnail Image in the file. In this case, the **Creator** SHOULD write the Thumbnail Image in the same orientation as the Primary Image. If the Thumbnail Image is not written with the same orientation, then the creator MUST include an appropriate orientation tag value in the thumbnail IFD.

A **Consumer** MAY choose to respect the orientation metadata included in a file when presenting an image or its thumbnail to the user. If a **Consumer** chooses to respect orientation metadata, it SHOULD:

- Treat the Primary Image orientation as Normal (value 1) if the Orientation tag of the Primary Image is missing.
- Treat the Thumbnail Image orientation as the same as the Primary Image if the Orientation tag of the Thumbnail Image is missing.

If a **Changer** alters the pixel content of the Primary Image, it SHOULD update or remove the Thumbnail Image (if previously present) so that a **Consumer** does not display an inappropriate thumbnail.

If a **Changer** alters the orientation metadata of the Primary Image, the **Changer** should also update the orientation metadata (if previously present) of the Thumbnail Image (if previously present) so that a **Consumer** does not display an inappropriate thumbnail.

Notes

The DCF specification states that a thumbnail MUST be stored in a fixed size of 160x120 pixels. The thumbnail MUST be cropped or padded with black to meet the 160x120 pixel size requirement regardless of the aspect ratio of the primary image.

Please consult the DCF specification for further details and restriction on JPEG images and thumbnails.

For more details such as type information and syntax, see the respective specifications.

5.5 Rating

The rating property allows the user to assign a fixed value (often displayed as “stars”) to an image. Usually, 1-5 star ratings are used. In addition, some tools support negative rating values (such as -1) that allows for marking “rejects” without deleting files in production.

Rating also can have fractional values. For example, online communities often deal with average values of rating coming from multiple users, which inevitably leads to fractional values.

Representation

Rating values are only available in XMP (xmp:Rating).

Type: Floating point number

Values: [-1.0; 0.0 … 5.0]

Guidance

The XMP (xmp:Rating) field SHOULD be read/written directly from/to the XMP.

Notes

The value -1.0 represents a “reject” rating.

If a client is not capable of handling float values, it SHOULD round to the closest integer for display and MUST only change the value once the user has changed the rating in the UI. Also, clients MAY store integer numbers.

If a value is out of the recommended scope it SHOULD be rounded to closest value. In particular, values > “5.0” SHOULD set to “5.0” as well as all values < “-1.0” SHOULD be set to “-1.0”.

5.6 Copyright

While, in the past, copyright information has principally been in the realm of the professional photographer, with the advent of easy online photo sharing sites, copyright has become increasingly important to the consumer as well. In the context of the consumer, this document focuses on two aspects:

- Copyright information
- URL to more information about the copyright

To avoid storing links as part of the copyright notice description, the optional copyright URL should be used to reference related information.

Representation

The CopyrightNotice information is available in the following properties:

Exif Copyright (33432, 0x8298)

IPTC CopyrightNotice (IIM 2:116, 0x0274)

XMP (dc:rights).

Guidance

Exif Copyright, IPTC CopyrightNotice, and XMP (dc:rights) are mapped together.

It is a good practice to include either the word “Copyright” or the copyright symbol “©” followed by a date and a descriptive reference to the user or company. Any further guidance on the definition of the copyright string is beyond the scope of this document. Please consult your national copyright office to learn more about the potential requirements for copyright statements in your area.

The CopyrightURL SHOULD be stored in XMP (xmpRights:WebStatement)

For more information see chapter “Handling Exif/TIFF, IPTC-IIM and XMP metadata” earlier in this document.

Restrictions

- Exif Copyright can contain 2 strings - creator and editor rights - separated by a nul (0x00) character.
- The length limitation in IPTC-IIM is 128 bytes.

Notes

The Exif Copyright information (creator and editor rights) MAY be concatenated by a linefeed character (0x0A) when stored in other formats.

For more details such as type information and syntax, see the respective specifications.

5.7 Creator

The creator, also known as "author", defines one or more creators of an image. Some cameras allow embedding creator information on image creation.

Representation

The creator is available in the following properties:

Exif Artist (315, 0x013B)

IPTC By-line (IIM 2:80, 0x0250)

XMP (dc:creator)

Guidance

The semicolon-space separator suggested by Exif SHOULD be recognized when mapping between the single TIFF Artist string and the individual array items in IIM By-line and XMP (dc:creator). If recognized, individual values in the TIFF Artist form MUST be enclosed in quotes (", U+0022) if they contain a semicolon-space or begin with a quote, and MAY be enclosed in quotes otherwise. Embedded quotes MUST be doubled when an individual value is enclosed in quotes.

For more information, see chapter "Handling Exif/TIFF, IPTC-IIM and XMP metadata" earlier in this document.

Restrictions

Length limitation in IPTC-IIM By-Line is a repeatable of 32 bytes each.

Notes

Exif Artist, IPTC By-line and XMP (dc:creator) are mapped together.

Individual names are separate items in the XMP (dc:creator) array as well as separate (repeated) IIM By-line tags.

Examples:

1. TIFF Artist from Exif specification:

```
Camera owner, John Smith; Photographer, Michael Brown; Image creator, Ken
James
```

Corresponding IPTC-IIM or XMP array items:

```
1. Camera owner, John Smith
2. Photographer, Michael Brown
3. Image creator, Ken James
```

2. TIFF Artist with quoted values:

```
first; with;semicolon; “with; semicolon-space”; “with; semicolon-space and
““quotes”““; non-leading “quotes”; “““leading”“ and non-leading ““quotes”““;
last
```

Corresponding IPTC-IIM or XMP array items:

```
1. first
2. with;semicolon
3. with; semicolon-space
4. with; semicolon-space and “quotes”
5. non-leading “quotes”
6. “leading” and non-leading “quotes”
7. last
```

For more details such as type information and syntax, see the respective specifications.

5.8 Location

The location of an image is one of the key pieces of information that consumers want to capture. Until recently location capture was often accomplished with post-creation keyword annotation. With the advent of embedded GPS, accurate location information can now be automatically inserted into image files at creation time. The Exif, IPTC-IIM, IPTC Core, IPTC Extension and XMP specifications all specify metadata properties that capture, with varying degrees of accuracy, either the location of the camera or the location of the image subject.

When storing location based information it's important to understand the difference between the two main concepts:

- "*Location Created*": This information describes the location where the image was created, the location of the camera during shot creation. The typical case is when a GPS receiver injects the current location into an image at shooting time (camera location).
- *"Location Shown*": This information describes the location of the main subject being shown in an image. For example, a picture of Mount Fuji would be tagged with the coordinates of where the mountain is located (subject location), although the picture may have been taken from downtown Tokyo.

In the latest specification (IPTC Extension 1.1), the IPTC has moved to a set of definitions that clearly differentiates between camera location and subject location defining both "*Location Created*" and "*Location Shown*" as a set of hierarchical properties (World Region, Country Name/ISO Country code, Province or State, City and Sublocation). The older [ambiguous] location definition (IPTC Core 1.0) is to be treated as legacy. This resolves any issues around the semantics of a textual location, and is clearly the way forward. Unlike keywords, which are unbounded, it is recommended that all location properties are being entered to form a valid hierarchy and avoid ambiguity (e.g. simply filling the City property as "Springfield", or State/Province as "Victoria" would represent multiple locations).

While the Exif specification allows for the capture of two sets of GPS location data, the Exif specification does not clearly state the purpose of the first set of coordinates vs. the second set (destination). This has led to a certain amount of confusion in the marketplace. In the consumer context, the camera location and subject location have often been treated the same. In case a semantic differentiation is made, it is very important to maintain these separate semantics. This specification will provide guidance for the creation of a clear, two part semantic (*Location Created* vs. *Location Shown*), the relationship of the GPS coordinate properties from the Exif specification and an algorithm for reconciling existing market implementations.

5.8.1 Location Created (GPS Latitude/Longitude)

Location information about where the image has been created.

Representation

GPS data: Exif GPS (34853:[1-6], 0x8825:[1-6]).

Guidance

The Exif GPS tags can be read directly without any reconciliation being required.

Notes

The Exif specification does not clearly state that the GPS tags 1 through 6 are to represent *Location Created*. These tags must be treated as such, with the understanding that there has been confusion in the marketplace and that reconciliation between *Location Created* and *Location Shown* may be required.

For more details such as type information and syntax, see the respective specifications.

5.8.2 Location Created (Textual Properties)

Textual location information about where the image was created.

Representation

Iptc4xmpExt:LocationCreated, a structure using the IPTC type *LocationDetails*.

Guidance

When a textual representation of the *Location Created* is to be included, an image **Creator** SHOULD use the IPTC Extension format representation. It MUST represent a real place which can be identified unambiguously without additional context. Here are some examples of valid and invalid places:

Example	Ambiguity	Validness
Grosse Elbstrasse 27, 22767 Hamburg, Germany	Unambiguous	Valid
“Jungle”	Ambiguous	Not valid
“My Backyard”	Only specific with additional context	Not valid

Places which cannot be identified unambiguously or without additional context SHOULD be treated as keywords.

When image **Changers** find legacy properties, they SHOULD be reconciled to the new hierarchical structure.

An image **Creator,** when adding textual location metadata, SHOULD do so using automated or validated entry methods.

To avoid the introduction of location ambiguity when using textual location properties, a **Creator** or **Changer** that adds these properties SHOULD add properties such that the hierarchy of location is complete to the lowest level entry added.

When there is a conflict between the Exif (GPS) *Location Created* and the textual representation, the Exif data MUST be considered correct.

Notes

The IPTC Extension specification introduced a mechanism that clearly defines the difference between where an image has been taken *Location Created* and where the content being shown on the image is located *Location Shown*. In both cases the location is represented as a five level hierarchical structure *Location Details* and is defined as follows:

- World Region
- Country Name and ISO-Code
- Province or State
- City
- Sublocation

It is possible to map Exif GPS data to these new properties. However, any transformation guidance is beyond the scope of the MWG guidelines at this time.

For more details such as type information and syntax, see the respective specifications.

5.8.3 Location Shown (GPS Latitude/Longitude)

Location information about the content being shown in the image.

Representation

GPS data: Exif GPS (34853:[19-26], 0x8825:[19-26]).

Guidance

The Exif GPS tags can be read directly without any reconciliation being required.

Notes

The Exif specification does not clearly state that the GPS tags 19 through 26 are to represent *Location Shown*. This has led to confusion in the marketplace mixing up *Location Created* with *Location Shown*. Moving forward, this differentiation should be clear but there may be a need to reconcile existing images according to the algorithm found in the chapter “Reconciling Location Data”.

5.8.4 Location Shown (Textual Properties)

Textual location information about the location being shown in the image.

Representation

Iptc4xmpExt:LocationShown, an unordered array of structures using the IPTC type *LocationDetails*.

Guidance

When a textual representation of the *Location Shown* is to be included, an image **Creator** SHOULD use the IPTC extension format representation. Same rules as applied to *Location Created* should be used.

Restrictions

According to the IPTC-IIM specification, the legacy Location, City and State text fields are limited to 32 bytes in length. Also, the Country text field is limited to 64 bytes in length. Any XMP based implementations such as IPTC Core or IPTC Extension do not have this limitation.

Notes

The following table shows the mapping from legacy IPTC-IIM and legacy IPTC Core properties to the new IPTC Extension Schema:

As noted in the guidance for *Location Created*, the new IPTC Extension for describing location, *Location Details*, is a structure, rather than discreet properties. Thus problems can arise when mapping the legacy properties to the new five level hierarchy if there are missing properties in that hierarchy.

Exif "*Destination*" only allows a single latitude/longitude pair to be stored in Exif GPS (34853:[19-26], 0x8825:[19-26]), while the textual representation of "*Location Shown*" is an XMP unordered array allowing multiple content locations to be recorded. Multiple "*Location Shown*" structures are acceptable, but the Exif Latitude/longitude pair is assumed to be pointing to the primary "*Location Shown*".

Location	IPTC-IIM	IPTC Core (legacy)	IPTC Extension (LocationDetails)
World Region			WorldRegion
Country	IIM 2:101;0x0265	photoshop:Country	CountryName
Province or State	IIM 2:95;0x025F	photoshop:State	ProvinceState
City	IIM 2:90;0x025A	photoshop:City	City

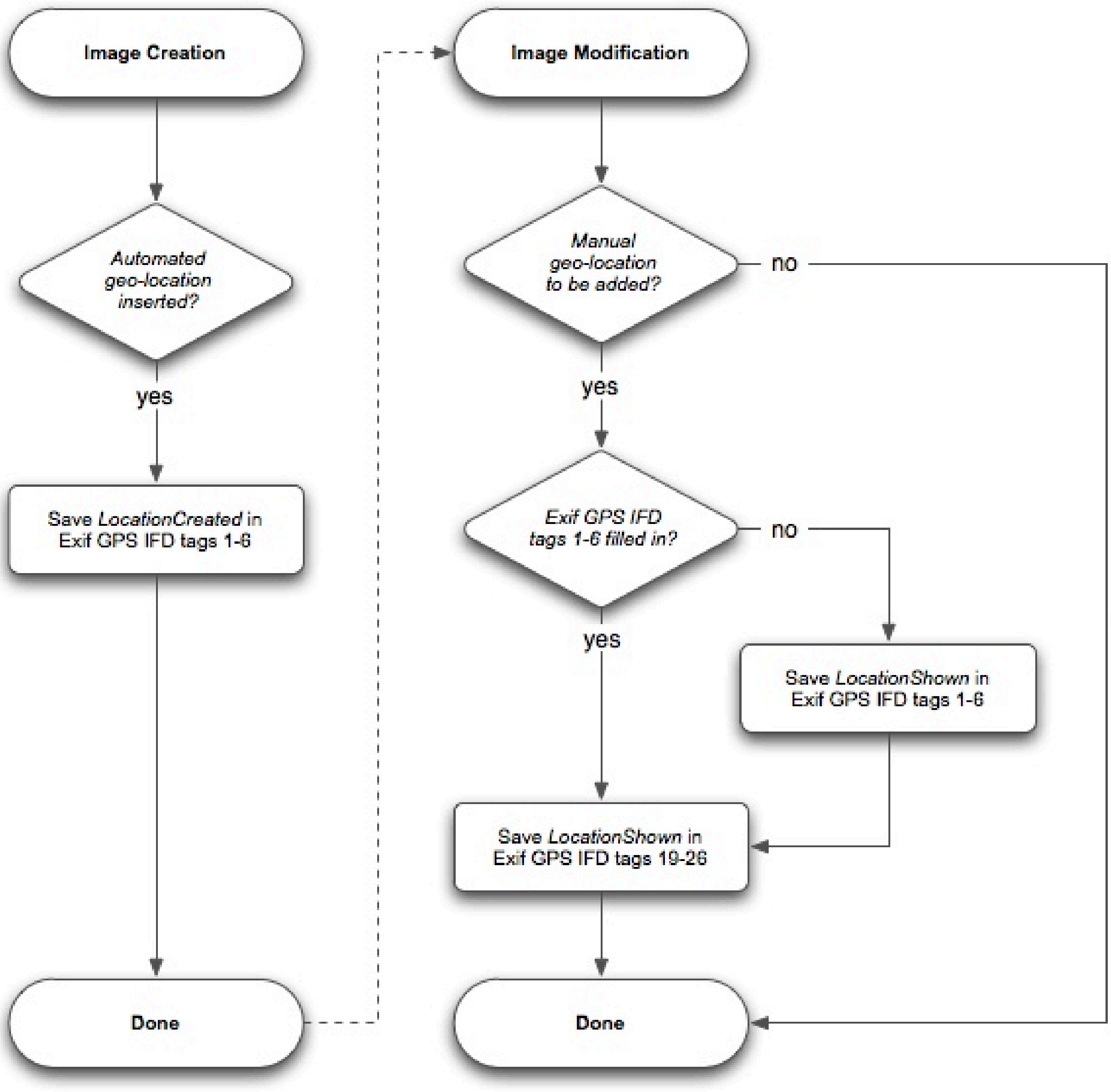
Sublocation	IIM 2:92;0x025C	Iptc4xmpCore:Location	Sublocation

5.8.5 Reconciling Location Data

Users, up until now, have not been given clear guidance for differentiating the location of the camera from the location of the subject of an image when tagging an image with a GPS location. The following algorithm will provide best-case guidelines for dealing with both new and existing image content, when annotating with GPS geo-location data.

As stated in the previous section Exif GPS (34853:[1-6], 0x8825:[1-6]) carries the camera location (*LocationCreated*) and Exif GPS (34853:[19-26], 0x8825:[19-26]) should be used to carry the GPS location of the subject of the image (*LocationShown*). In the vast majority of cases, when an image is tagged with the camera location, the tagging is done through automated means. Today all GPS tagging is inserted in locations 1-6 of the Exif GPS IFD. Problems may occur when GPS tagging occurs after photo creation, and is done in a more manual way. Generally, post-shot annotation is concerned with the location of the subject, as the location of the camera may not be known. The following flow chart will maximize both compatibility with the existing marketplace realities and maximize the likelihood that annotations will be correctly interpreted by conforming applications.

See the following diagram for an overview:



5.9 Image Region Metadata

Today metadata in image files is applied to the entire content of the image. This is the first step in a consumer's image organization. With recent advances in camera technology, such as in-camera face detection or focus point selection, and the increased popularity of various social networking and photo sharing sites, applying metadata to specific regions within an image has become more important. Specifically, popular social networking sites allow users to add keywords to user-defined regions within an image. Standardizing a way for this metadata to be preserved in the file is crucial to the consumer's workflow of image organization.

With metadata applied to specific regions, certain requirements are necessary so that these regions may be utilized in meaningful ways. First, and foremost, a list of regions must be captured. Metadata per region may be different, so it is also necessary to allow for different types of information to be stored. It is critical to remember that a variety of image transformations, such as rotation, resizing, or cropping, may invalidate the region that was applied to the actual image bits stored within the file.

Technical Issues

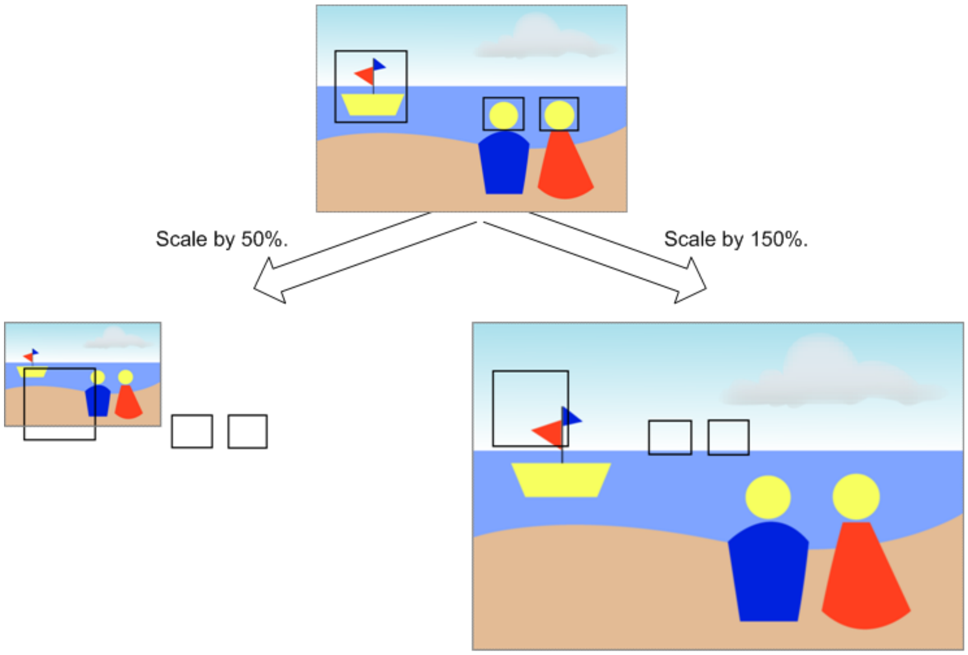
Once an image is taken, during its lifetime many actors will operate on that image and its metadata. Because of this, some issues may arise with rationalizing the region metadata stored within the image. When non-compliant actors have processed the image, the integrity of the region metadata may be invalidated. It is important that compliant actors validate region metadata as well as update it such that the region metadata becomes valid.

If an actor does not properly update the region metadata, issues will arise when certain operations are performed on an image. We will discuss the issues that operations such as resizing, rotation, and cropping have on the region metadata and what can be done to preserve the integrity of that metadata. This discussion covers general issues regarding the storage of image coordinates within metadata. However, it is important to remember that there are a vast number of other morphological operations that transform image data in ways where previously stored region metadata may be affected and made invalid. Examples of such operations include straightening, correcting for lens distortions, skewing, as well as some artistic effects. Typically any operation that operates by changing the mapping of pixels between the input and output will cause issues with region metadata. It is important to remember that for all morphological operations, region metadata has to be updated either manually or algorithmically so that the integrity of this metadata is preserved.

The following section examines issues which arise when the operations of resizing, rotating, and cropping are performed on an image. Other morphological operations will not be discussed.

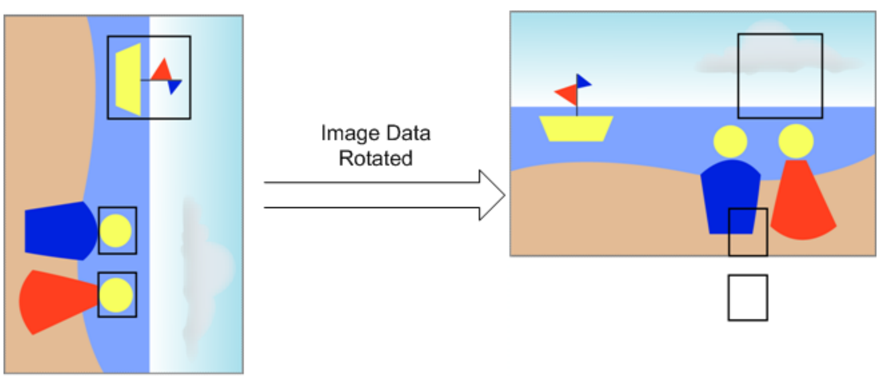
Resize Issues

A common problem with storing region coordinates is that when an image is resized, the coordinates are no longer applicable to the originally tagged regions. In the case of downsizing the image, the original region's coordinates may no longer be within the confines of the image dimensions. In the case of upsizing, the original region's coordinates are within the new image dimensions however the sub-images represented by each region are no longer the same. When resizing images, the coordinates of the regions must be updated to reflect the where the region that was originally identified has moved to. In order to prevent these problems, the use of normalized coordinates is encouraged (i.e. ones where the value is represented as percentage of the width and height, within the range of [0…1]).



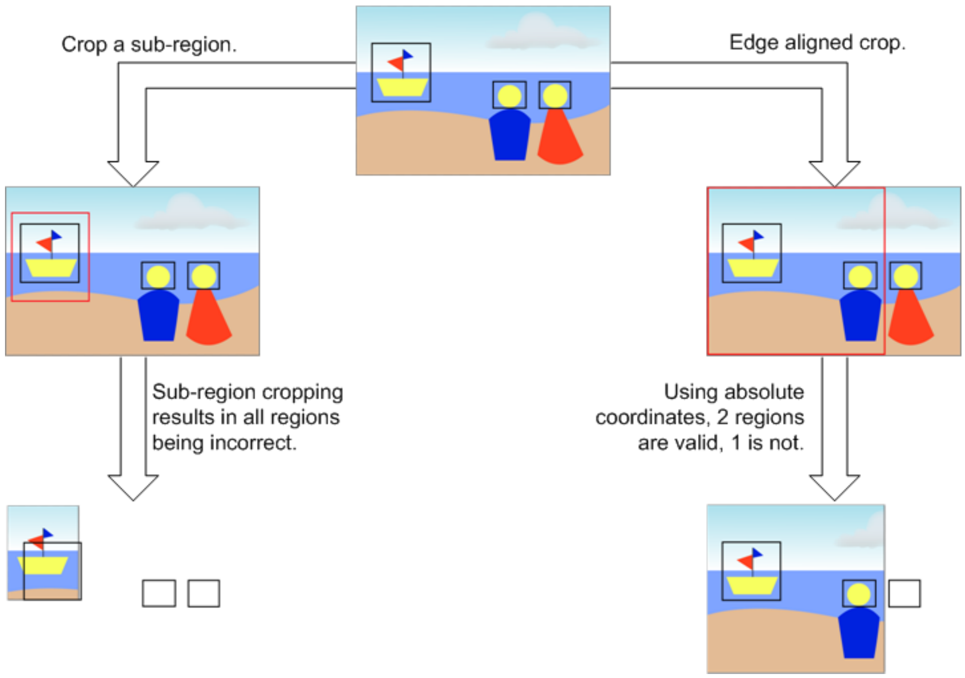
Rotation Issues

As discussed earlier in section 5.4, regarding Exif Orientation, if metadata is not updated upon a physical rotation of the image data some applications may incorrectly display the image. Region metadata is applied to the stored image. When applying a rotation by applying Exif Orientation, the rotation must be applied to the regions as well. Otherwise, the regions will not apply to the areas intended. If the stored image is rotated, the region metadata must also be updated to reflect the change.



Cropping Issues

During an image crop, a portion of the image is being discarded. Due to this, if the image had any region metadata in it, those regions may no longer be applicable. Specifically, there are three cases whereby region metadata needs to be updated. In the first case, if a crop is performed around a specific object, then certain regions will be completely lost and even the region, which closely matched the crop, will need to be redefined. The second case exists when a portion of the image is cropped out, where some of the regions will no longer be within the new coordinates of the image. In the third case, if a crop is performed through an existing region, then as long as the center of the region exists in the cropped image, this region should be retained. In all of these cases, region metadata must be updated to reflect the changes made to the image.



Representation

The field namespace is http://www.metadataworkinggroup.com/schemas/regions/

The preferred prefix is *mwg-rs*.

Regions

Field Name	Value Type	Description
mwg-rs:Regions	RegionInfo	Main structure containing region based information.

RegionInfo

Field Name	Value Type	Description
mwg-rs:RegionList	Bag of RegionStruct	List of Region structures.
mwg-rs:AppliedToDimensions	Dimensions	Width and height of the image at the time of

		processing when storing image region metadata.

RegionStruct

Field Name	Required	Value Type	Description
mwg-rs:Area	Required	Area	Area that describes the region.[3]
mwg-rs:Type	Optional	Closed Choice	Type purpose of the region: • Face • Pet • Focus • BarCode
mwg-rs:Name	Optional	Text	Name or short description of content shown in image region. For example, name of person or pet.
mwg-rs:Description	Optional	Text	Descriptive text for content shown in image region.
mwg-rs:FocusUsage	Required if mwg-rs:Type = "Focus"	Closed Choice	Usage scenario for a given focus area. Three different types exist: ▪ EvaluatedUsed ▪ EvaluatedNotUsed ▪ NotEvaluatedNotUsed
mwg-rs:BarCodeValue	Optional	Text	Decoded BarCode value string.
mwg-rs:Extensions	Optional	Undefined Struct	Any additional top level XMP property to describe the content shown in image region.

For more information about the intended usage of region types and related properties, please see region type table and samples below.

[3] *See Appendix C at the end of this document for more information.*

Guidance

Region Coordinates

A **Creator** or **Changer** MUST store the region as a point, circle, or rectangle. This is stored as the center point, along with a diameter or width and height to match the point, circle or rectangle. The center MUST be stored as a normalized value where its range is [0…1], indicating that it is located within the bounds of the image. The width, height, and diameter MUST be stored as normalized, positive values. With such a range it is possible to represent a region where a portion of it is outside the visible bounds of the image. For any region which is a circle, the diameter MUST be relative to the smaller value of the image width or height.

Exif Orientation also affects the interpretation of height and width. A **Creator** or **Changer** MUST express region coordinates, width and height relative to the stored image, prior to the application of the Exif Orientation tag. In other words, the origin of the image is the upper left. This allows the region information to remain stable when Exif Orientation is taken into account.

A **Consumer** MUST respect region metadata when the center point is within the bounds of [0..1] and the dimensions of the rectangle or circle are also within those bounds. When the center point is outside of the bounds of [0..1], a **Consumer** SHOULD ignore these regions. In the case where the center point is inside of the bounds of [0..1], but the dimensions of the rectangle or circle extend beyond those bounds, a **Consumer** SHOULD clip to the bounds of the image, rather than ignore the region metadata.

"Applied To" State

Since the file may be modified by actors that are either non-compliant to this guidance or choose not to participate in image region metadata, it is critical to retain information about what the state of the image was when the region was specified. A **Creator** MUST store the region "applied to" width and height.

A **Changer** MUST update the "applied to" value upon a resize, crop, or physical rotation in order to ensure that the region information is still valid for the new state of the file. In the case of a crop or physical rotation, a **Changer** MUST also update the region coordinates so as to further ensure that the region information remains valid. A **Changer** SHOULD remove region information for regions that are no longer applicable (e.g. a region that was orphaned due to a crop or rotation operation).

A **Consumer** MUST validate the current state of the image and compare this against the "applied to" state to ensure that the region information is up-to-date. The information specified in the region MUST be validated to ensure that the data is applicable to the current state of the file. In the case where the "applied to" size and actual image size differ, if the ratio is significantly different a **Consumer** SHOULD ignore the region. Otherwise a **Consumer** or **Changer** MAY update the region metadata as though the image were scaled to the new dimensions.

Region Content Type

The region type is optional. If the content identified in a region is of one of the types specified, a **Creator** or **Changer** MUST specify the type of the region. If the content region identified in a region is not one of the types specified, a **Creator** or **Changer** MUST NOT write this type information. A **Consumer** MAY utilize this type information to perform specialized actions.

See the following table for an overview of type definitions:

Type	Description
Face	Region area for people's faces.
Pet	Region area for pets.
Focus	Region area for camera auto-focus regions. Three different usage types of focus points exist: *EvaluatedUsed* specifies that the focus point was considered during focusing and was used in the final image. *EvaluatedNotUsed* specifies that the focus point was considered during focusing but not utilized in the final image. *NotEvaluatedNotUsed* specifies that a focus point was not evaluated and not used, e.g. a fixed focus point on the camera which was not used in any fashion. Focus usage types MUST be stored in mwg-rs:FocusUsage. If no usage type is available the image region MUST NOT be written.
BarCode	One dimensional linear or two dimensional matrix optical code. Any decoded barcode value MAY be stored in mwg-rs:BarCodeValue. This property allows carrying any decoded barcode value such as, for example, an ISBN number. However, this field does not define any further semantic meaning so it is the responsibility of the **Consumer** to put this into the user's context.

Region Metadata

Metadata applied to a region MUST be done in one of four ways.

1. Use the top level properties such as mwg-rs:Name or mwg-rs:Description.
2. Refer to XMP metadata already present using the rdfs:seeAlso notation. With this approach, any metadata within XMP may be referred to. For example, if IPTC Extension 1.1 has previously been used to identify who was in the photo, a region may refer to the Iptc4XmpExt:PersonInImage struct.
3. Nest any top level XMP metadata within mwg-rs:Extensions. With this approach additional or arbitrary metadata can be applied locally to the region.
4. Use a combination of (1), (2) or (3). In this manner, pre-existing XMP metadata is reused and new metadata is applied directly to the region.

Samples

The following chapter contains some image region examples. Namespace definition and top level RDF properties have been ignored for improved readability.

Simple image region samples

```
...
<mwg-rs:Regions rdf:parseType="Resource">
  <mwg-rs:AppliedToDimensions stDim:w="4288" stDim:h="2848" stDim:unit="pixel"/>
  <mwg-rs:RegionList>
    <rdf:Bag>
      <!-- Simple example for face detection -->
      <rdf:li rdf:parseType="Resource">
        <mwg-rs:Area stArea:x="0.5" stArea:y="0.5" stArea:w="0.06" stArea:h="0.09" stArea:unit="normalized"/>
        <mwg-rs:Type>Face</mwg-rs:Type>
        <mwg-rs:Title>John Doe</mwg-rs:Title>
      </rdf:li>
      <!-- Simple example for pet detection -->
      <rdf:li rdf:parseType="Resource">
        <mwg-rs:Area stArea:x="0.5" stArea:y="0.5" stArea:w="0.06" stArea:h="0.09" stArea:unit="normalized"/>
        <mwg-rs:Type>Pet</mwg-rs:Type>
        <mwg-rs:Title>Fido</mwg-rs:Title>
        <mwg-rs:Description>Fido looks happy!</mwg-rs:Description>
      </rdf:li>
      <!-- Metadata applied to a specific region, defined locally -->
      <rdf:li rdf:parseType="Resource">
        <mwg-rs:Area stArea:x="0.5" stArea:y="0.5" stArea:w="0.003" stArea:h="0.002" stArea:unit="normalized"/>
        <mwg-rs:Type>Focus</mwg-rs:Type>
        <mwg-rs:FocusUsage>EvaluatedUsed</mwg-rs:FocusUsage>
      </rdf:li>
      <rdf:li rdf:parseType="Resource">
        <mwg-rs:Area stArea:x="0.5" stArea:y="0.5" stArea:w="0.003" stArea:h="0.002" stArea:unit="normalized"/>
        <mwg-rs:Type>BarCode</mwg-rs:Type>
        <mwg-rs:BarCodeValue>ISBN:1234567890</mwg-rs:BarCodeValue>
        <mwg-rs:Name>The Best Book</mwg-rs:Name>
        <mwg-rs:Description>The best book is the best book ever.</mwg-rs:Description>
      </rdf:li>
    </rdf:Bag>
  </mwg-rs:RegionList>
</mwg-rs:Regions>
...
```

Reference to other metadata inside the same XMP packet

```
...
<Iptc4xmpExt:PersonInImage>
    <rdf:Bag>
      <rdf:li>David Smith</rdf:li>
    </rdf:Bag>
</Iptc4xmpExt:PersonInImage>

...
<mwg-rs:Regions rdf:parseType="Resource">
  <mwg-rs:AppliedToDimensions stDim:w="4288" stDim:h="2848" stDim:unit="pixel"/>
  <mwg-rs:RegionList>
    <rdf:Bag>
      <!-- Metadata applied to a specific region, defined elsewhere -->
      <rdf:li rdf:parseType="Resource">
        <mwg-rs:Area stArea:x="0.5" stArea:y="0.5" stArea:w="0.06" stArea:h="0.09" stArea:unit="normalized"/>
        <mwg-rs:Type>Face</mwg-rs:Type>
        <rdfs:seeAlso rdf:resource="Iptc4xmpExt:PersonInImage"/>
      </rdf:li>
    </rdf:Bag>
  </mwg-rs:RegionList>
</mwg-rs:Regions>
...
```

Combination of reference and local metadata

```
...
<dc:subject>
  <rdf:Bag>
    <rdf:li>Felix</rdf:li>
  </rdf:Bag>
</dc:subject>
...
<mwg-rs:Regions rdf:parseType="Resource">
  <mwg-rs:AppliedToDimensions stDim:w="4288" stDim:h="2848" stDim:unit="pixel"/>
  <mwg-rs:RegionList>
    <rdf:Bag>
      <!-- Metadata applied to a specific region, different metadata defined locally and elsewhere -->
      <rdf:li rdf:parseType="Resource">
        <mwg-rs:Area stArea:x="0.5" stArea:y="0.5" stArea:w="0.06" stArea:h="0.09" stArea:unit="normalized"/>
        <mwg-rs:Type>Pet</mwg-rs:Type>
        <mwg-rs:Title>Felix</mwg-rs:Title>
        <mwg-rs:Description>Felix is a famous cat!</mwg-rs:Description>
```

```
          <!-- Link to keywords -->
          <rdfs:seeAlso rdf:resource="dc:subject"/>
          <!-- Additional information -->
          <mwg-rs:Extensions rdf:parseType="Resource">
            <xmpRights:UsageTerms>
              <rdf:Alt>
                <rdf:li xml:lang="x-default">Do not use any image of this famous cat for advertisement!</rdf:li>
              </rdf:Alt>
            </xmpRights:UsageTerms>
          </mwg-rs:Extensions>
        </rdf:li>
      </rdf:Bag>
    </mwg-rs:RegionList>
  </mwg-rs:Regions>
  ...
```

Some additional samples

```
  ...
  <mwg-rs:Regions rdf:parseType="Resource">
    <mwg-rs:AppliedToDimensions stDim:w="4288" stDim:h="2848" stDim:unit="pixel"/>
    <mwg-rs:RegionList>
      <rdf:Bag>
        <!-- Simple comment for an arbitrarily user defined selection with unkown type -->
        <rdf:li rdf:parseType="Resource">
          <mwg-rs:Area stArea:x="0.5" stArea:y="0.5" stArea:w="0.06" stArea:h="0.09" stArea:unit="normalized"/>
          <mwg-rs:Description>This is a beautiful mountain!</mwg-rs:Description>
        </rdf:li>
        <!-- More complex attribution example with author information -->
        <rdf:li rdf:parseType="Resource">
          <mwg-rs:Area stArea:x="0.5" stArea:y="0.5" stArea:w="0.06" stArea:h="0.09" stArea:unit="normalized"/>
          <mwg-rs:Description>This comment has been added by John.</mwg-rs:Description>
          <mwg-rs:Extensions rdf:parseType="Resource">
            <dc:creator>
              <rdf:Seq>
                <rdf:li>John Doe</rdf:li>
              </rdf:Seq>
            </dc:creator>
          </mwg-rs:Extensions>
        </rdf:li>
      </rdf:Bag>
    </mwg-rs:RegionList>
  </mwg-rs:Regions>
```

5.10 Hierarchical Keywords

Keywords are one of the most popular and effective tagging mechanisms for images and were discussed at length in the first version of the Metadata Working Group's (MWG) guidance, published in 2008. However, with a growing number of keywords in a given portfolio, users are looking for smarter ways to organize and assign keywords while making sure they still can filter and search for their images effectively. Rich tagging (hierarchies, synonyms, and external vocabularies) is not only essential to the professional photographer, but is also of increasing value to the private photographer with a large portfolio or using Internet photo sharing.

Most popular image management applications support a notion of keyword hierarchies.

Some common characteristics of the user models are:

- Tree structured hierarchies created by the user
- Any node (word) in the tree may be assigned to a file
- Homonyms are properly supported
- Searching / filtering implicitly includes lower level words

In addition, some applications support:

- Import and export of vocabulary files to preset keyword user interface elements
- Synonyms - related terms describing the keyword for more effective searching
- Categories - grouping of keywords that is only used for organizational purposes

The goal for the Metadata Working Group (MWG) is to define a hierarchical keyword policy that preserves important existing behavior, promotes better future behavior and is pragmatic enough to be adopted. Based on a sample hierarchy, this chapter will discuss the different topics of advanced keywording such as synonyms, categories and vocabularies. Subsequently, concrete guidance will be given on how to store hierarchical keywords in an image file. The chapter will close with an overview of existing solutions and compatibility considerations.

Hierarchies

Hierarchies are used for organizing and assigning keywords. Hierarchical keywords simply provide the syntactic mechanism for relating one keyword to another without any implied relationship to a global context, such as an external vocabulary or asset management system. The assumption is that the hierarchies are structured from higher to lower level whereas each hierarchy node - keyword within themselves - can be assigned to an image individually. Assigning a specific node implies that all parent nodes will be adequate keywords for this image as well. Searching for a given node implicitly includes lower level keywords assigned to this file.

Here are three simple hierarchies that will be used here to illustrate the user and embedded metadata models.

- Places
    - States
        - Georgia
        - Wyoming
- People
    - Georgia
    - Cat
- Animals
    - Mammals
        - Cat
        - Dog

The uses of “Georgia” are homonyms, distinct keywords with the same spelling. Proper support for hierarchical keywords requires knowing which “Georgia” is attached to a file, and not mixing search results.

Synonyms

The value of synonyms is to automatically extend search results when importing or exporting files, while not forcing the photographer to manually apply all synonyms. For example, “Cats”, “Feline”, and “Felines” might be defined as synonyms of “Cat” the “Mammal”. As an extension of homonyms, proper support of synonyms differentiates synonyms of “Cat” the “Mammal” from synonyms of “Cat” the “Person”.

Note: This MWG guidance will not provide a specific solution for synonyms, as it seems not worth the effort to introduce this level of complexity for the consumer.

Categories

The perceived motivation for categories is to have nodes in the hierarchy that serve only to help organize the keywords. The applications that support categories (Adobe Lightroom and Photo Mechanic) do so by allowing any node to be called a category instead of a normal keyword. For example, “States” might be called a category. In that case, searching for “States” might not be allowed and metadata embedded in a file might only mention “Places” and “Wyoming”, leaving out “States”.

Note: This MWG guidance will not provide a specific solution for categories, as it seems not worth the effort to introduce this level of complexity for the consumer.

Representation

The following guidance only needs to be followed if an actor is supporting hierarchical keywords. In other words, it's not necessary to support hierarchical keywords to be MWG compliant.

The field namespace is http://www.metadataworkinggroup.com/schemas/keywords/

The preferred prefix is *mwg-kw.*

Keywords

Field Name	Value Type	Description
mwg-kw:Keywords	KeywordInfo	Main structure containing keyword based information. *Note: The main intent to define a top level structure is that it could be used to host other properties such as references to external catalogues in the future.*

KeywordInfo

Field Name	Value Type	Description
mwg-kw:Hierarchy	Bag of KeywordStruct	List of root keyword structures.

KeywordStruct

Field Name	**Value Type**	**Description**
mwg-kw:Keyword	Text	Name of keyword (-node).
mwg-kw:Applied	Boolean	*True* if this keyword has been applied, *False* otherwise. If missing, mwg-kw:Applied is presumed *True* for leaf nodes and *False* for ancestor nodes.
mwg-kw:Children	Bag of KeywordStruct	List of children Keywords structs.

Guidance

A **Consumer** …

- MUST read the XMP “Hierarchy” and “Keyword” elements to construct the keyword hierarchy.
- MUST preserve all nested information - even if it's no longer used. This includes unknown elements.
- MAY interpret nested information like the XMP “Applied” property for further use.

A **Changer** …

- MUST consume the information according to the reconciliation guidance described above and, in particular, MUST preserve all nested information even if unknown.
- MUST write the XMP dc:subject property to store the individual keywords. Hierarchical path elements MUST be flattened, which means that each hierarchy node needs to be stored as a separate keyword entry to XMP dc:subject.
- MUST write the XMP “Hierarchy” and “Keyword” elements as described below.

SHOULD write the XMP “Applied” element, indicating that the user had selected a particular element.

Sample

The following simple catalog structure will illustrate the serialization (the star indicates the specific keyword node has been selected by the user):

Animals

--- Mammals*

--- Cat*

--- Dog

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description xmlns:dc="http://purl.org/dc/elements/1.1/">
    <!-- flat keyword list for interoperability -->
    <dc:subject>
      <rdf:Bag>
        <rdf:li>Animals</rdf:li>
        <rdf:li>Mammals</rdf:li>
        <rdf:li>Cat</rdf:li>
      </rdf:Bag>
    </dc:subject>
  </rdf:Description>

  <rdf:Description xmlns:mwg-kw="http://www.metadataworkinggroup.com/schemas/keywords/">
    <mwg-kw:Keywords rdf:parseType="Resource">
      <!-- hierarchy definitions -->
      <mwg-kw:Hierarchy>
        <rdf:Bag>
          <!-- first level -->
          <rdf:li rdf:parseType="Resource">
            <mwg-kw:Keyword>Animals</mwg-kw:Keyword>
            <mwg-kw:Applied>False</mwg-kw:Applied>
            <mwg-kw:Children>
              <rdf:Bag>
                <!-- second level -->
                <rdf:li rdf:parseType="Resource">
                  <mwg-kw:Keyword>Mammals</mwg-kw:Keyword>
                  <mwg-kw:Applied>True</mwg-kw:Applied>
                  <mwg-kw:Children>
                    <rdf:Bag>
                      <!-- third level -->
                      <rdf:li rdf:parseType="Resource">
                          <mwg-kw:Keyword>Cat</mwg-kw:Keyword>
```

```
                          <mwg-kw:Applied>True</mwg-kw:Applied>
                      </rdf:li>
                    </rdf:Bag>
                  </mwg-kw:Children>
                </rdf:li>
              </rdf:Bag>
            </mwg-kw:Children>
          </rdf:li>
        </rdf:Bag>
      </mwg-kw:Hierarchy>
    </mwg-kw:Keywords>
  </rdf:Description>
</rdf:RDF>
```

Note: The keyword “Dog” has not been serialized, as it is a leaf node and neither required to reconstruct the keyword hierarchy nor any other information.

5.11 Collections

Images are organized and classified by people for diverse reasons. Users may want to group files together by content, type, time, or any arbitrary criteria. Often, an image file’s classification system can represent meaningful hints as to how the image should be viewed and managed.

Professional digital asset management (DAM) systems manage images with highly structured hierarchies and vocabularies for searching and accessing content. However, consumers need to organize and search their images using simple methods, in particular:

- Consumers are generally unwilling to invest large amounts of resources in creating and maintaining classification hierarchies.
- Consumer semantics are unbounded, because motivations driving user classification strategies are unique and organizational requirements often change over time.

While there is an unbounded set of reasons and mechanisms consumers might have for creating particular groupings of images, a number of common cases emerge:

- “My Favorite Flower Shots”, “Family Photos”
  - ➢ Content-based classification *[who, what, where]*
- “Summer 2009 Photos”, “DSC-231 - DSC-554”
  - ➢ Time-based classification *[when]*
- “http://www.flickr.com/photos/luigi-san“, “Scans from Grandpa’s photos”
  - ➢ Resource/process-based grouping (e.g., stored in an Internet repository, processed in a certain way) *[how]*
- “Bobby’s photos”, “http://www.istockphoto.com/file_closeups.php“
  - ➢ Ownership or user-based classification

A common way for consumers to present images to a wide audience is through publishing image galleries on the Internet. This implies that there is a resource available for the description, organization and / or publication of a set of images. A useful descriptor for a gallery of images on the Internet is the URI which identifies that gallery. It acts as a "breadcrumb" leading back to the source where an image is stored and managed as a part of a larger context. Collections may be hierarchical in nature, and URIs (or extended naming conventions) can provide a natural way to represent such a hierarchy. However, the organizational structure of a collection pointed to by the collection metadata property is out of scope. While the title of the gallery may hint at the common collection semantic, full intent is often only understood within the context of the storage and management application that contains it.

Representation

Collections may be identified by either a name or a URI or a combination of both. The URI, if present, uniquely identifies the collection location, whereupon the name is advisory and presentational. Otherwise, the name identifies the collection. A string match of the CollectionName alone cannot guarantee global uniqueness, as multiple collections may have the same name. Images may be a member of zero or more collections.

The field namespace is http://www.metadataworkinggroup.com/schemas/collections/

The preferred prefix is *mwg-coll.*

Collections

Field Name	Value Type	Description
mwg-coll:Collections	Bag of CollectionInfo	List of collection structures.

CollectionInfo

Field Name	Value Type	Description
mwg-coll:CollectionName	Text	Textual name of the Collection to which this image is a member. No specific criteria or mechanism is implied for membership.
mwg-coll:CollectionURI	URI	URI describing the collection resource.

Guidance

A **Creator**…

- A **Creator** MUST either write CollectionName or CollectionURI or a combination of both. The CollectionURI MUST conform to the URI syntax.

A **Consumer** …

- MUST treat the CollectionName as a string. No implied formatting of that string can be assumed. Hence, any interpretation of the CollectionName is a subjective (non-machine) activity.
- SHOULD treat the CollectionURI as an Internet identifier. As such it does not guarantee that there exists an Internet resource in which a copy of this media can be found.

A **Changer**…

- MUST consume the information according to the reconciliation guidance described above.
- SHOULD NOT remove any existing collection properties. Any added collection semantics added MUST meet the above requirements for **Creators**.

Sample

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description xmlns:mwg-coll="http://www.metadataworkinggroup.com/schemas/collections/">
    <mwg-coll:Collections>
      <rdf:Bag>
        <!-- list of collections to which this media belongs -->
        <rdf:li rdf:parseType="Resource">
          <mwg-coll:CollectionName>My Last Vacation Photos</mwg-coll:CollectionName>
          <mwg-coll:CollectionURI>http://www.flickr.com/photos/myvacation</mwg-coll:CollectionURI>
        </rdf:li>
        <rdf:li rdf:parseType="Resource">
          <mwg-coll:CollectionName>Beautiful Sunset Set</mwg-coll:CollectionName>
        </rdf:li>
      </rdf:Bag>
    </mwg-coll:Collections>
  </rdf:Description>
</rdf:RDF>
```

APPENDIX A: REFERENCES

Metadata Standards

Exif	http://www.jeita.or.jp / http://www.cipa.jp
IPTC	http://www.iptc.org
IPTC-IIM	http://www.iptc.org/IIM
IPTC Core for XMP	http://www.iptc.org/photometadata
IPTC Extension for XMP	http://www.iptc.org/photometadata
XMP	http://www.adobe.com/products/xmp

Metadata Specifications

Exif 2.3

http://www.cipa.jp/english/hyoujunka/kikaku/pdf/DC-008-2010_E.pdf

DCF 2.0 (2010)

http://www.cipa.jp/english/hyoujunka/kikaku/pdf/DC-009-2010_E.pdf

IPTC-IIM 4.1

http://www.iptc.org/std/IIM/4.1/specification/IIMV4.1.pdf

IPTC Core 1.0

http://www.iptc.org/std/Iptc4xmpCore/1.0/specification/Iptc4xmpCore_1.0-spec-XMPSchema_8.pdf

IPTC Core 1.1 & IPTC Extension 1.1

http://www.iptc.org/std/photometadata/specification/IPTC-PhotoMetadata-201007.pdf

XMP

http://www.adobe.com/devnet/xmp/

File Format Specifications

JPEG

http://www.jpeg.org/jpeg/

TIFF

http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf

PSD/PSIRs

http://www.adobe.com/go/psir

Miscellaneous

RDF

http://www.w3.org/TR/rdf-schema

Dublin Core

http://dublincore.org/documents/dces

RFC2119

http://www.ietf.org/rfc/rfc2119.txt

Date and Time (W3C)

http://www.w3.org/TR/NOTE-datetime

RFC 3986 - URI Definition

http://www.ietf.org/rfc/rfc3986.txt

APPENDIX B: IMPLEMENTATION NOTES

This appendix provides non-normative notes to help implementers follow the letter and spirit of the formal guidelines. The words “may”, “should”, and “must” are used here in lower case, with their usual informal meanings.

Policy for Creators and Changers

The language of sections 3.1.1 and 3.1.2 has some subtleties that might not be noticed on a first reading. The basic intent is that Creators need to write compliant metadata that they know to be correct, and Changers need to strive to preserve information while ensuring that changes leave the metadata relevant and consistent.

The last sentence of the first paragraph of section 3.1.1 is specifically meant as a “loophole” for editing applications that wish to be strict about their output: “Alternately, an image editing application might behave as a creator even though it produces a new file from an existing file.” This is saying that it is a legitimate design choice for an editing application to behave as a creator of its output, rather than a changer of its input. Users will learn that and not be surprised.

In section 3.1.2, “deletion” really refers to “total elimination”, or deletion of all forms. As one of the examples cites, it is okay to remove some forms while keeping others. Doing that might have broader compatibility effects, e.g. making the metadata hidden from readers of only the deleted forms. But such readers are by definition not MWG-compliant; an MWG-compliant writer is not required to cater to non-compliant readers.

The phrases “done with specific intent” or “by explicit user intent” are intentionally vague. There can be many kinds of intent. For example, a user might ask that a saved file be as small as possible, or be redacted for publication. It is a legitimate design choice for this to involve deletion of metadata. Ideally additional safeguards should be placed around sensitive metadata such as a copyright, but that is the purview of the application, not the MWG.

Forms of metadata written by Creators and Changers

There is intentionally no specific guidance about which forms of metadata should be written by Creators and Changers. This of course only applies to those items that can appear in more than one form. Compliant readers are intentionally required to look for all forms, and to reconcile among multiple forms. This gives writers the freedom to write whatever forms they find convenient, so long as those forms in the file are consistent.

Of course the broadest compatibility among MWG-compliant and non-compliant readers will be attained by writing all forms. But it is not up to the MWG to advocate the value of that over other considerations.

Local encoding of text in Exif and IPTC-IIM

Several sections mention an undefined, non-Unicode, “local” encoding for text. Some applications, especially those from Adobe, utilize the Windows and Macintosh notions of a “current default” character encoding for this. This encoding typically relates to a user's choice of UI language, and can be modified on any machine at any time.

This is not always code page 1252 for Windows and MacRoman on Macintosh, those values typically apply only to the United States and portions of Western Europe. For Windows, this interpretation of local encoding is CP_ACP. For Macintosh, this is based on smSystemScript.

Internal and external modification times

Developers can understandably get confused between the internal modification time of Exif DateTime or XMP xmp:ModifyDate, and the external file modification time maintained by the file system. There is no intent that these be identical.

The intent is that software set the internal modification time to the current local time at some convenient point when writing a file. Preferably, this should not be significantly before closing the file, and the internal and external times will often be close as a consequence of this. The mention of "current local time" is intentional, it is reasonable and appropriate for xmp:ModifyDate to contain the local time zone offset.

Note that the external file system time can be significantly different, for example if the file is on an external server in another time zone. This difference and the possible "reset" of the external modification time if a file is copied are some of the rationale for having an internal modification time.

No specific guidance is given for the choice of displaying, sorting, etc., based on the internal or external modification time. That should be appropriate to the context, aiming to give users the most value and least confusion. When only looking at images, it seems reasonable to use the internal image modification time. When looking at files in general, it seems reasonable to use the file system time.

APPENDIX C: XMP DEFINITIONS

This chapter contains missing XMP definitions that are necessary to support the new schemas.

Area

This structure represents an area. Similar to Dimensions (stDim) the “unit” field describes the specific unit being used. This is based on the definition of Exif SubjectArea. The following table gives an overview of the different types and properties being used.

Type	Definition
Point	Single point at stArea:x, stArea:y
Circle	Center at stArea:x, stArea:y with diameter stArea:d
Rectangle	Center at stArea:x, stArea:y with bounds stArea:w, stArea:h

The field namespace URI is http://ns.adobe.com/xmp/sType/Area#

The preferred field namespace prefix is stArea

Field Name	Value Type	Description
stArea:x	Real	X coordinate of the center of the area (point, circle, rectangle)
stArea:y	Real	Y coordinate of the center of the area (point, circle, rectangle)
stArea:w	Real	Width of the area (rectangle)
stArea:h	Real	Height of the area (rectangle)
stArea:d	Real	Diameter of area (circle)
stArea:unit	Open Choice	In the context of this document, only “normalized” is being specified for handling image regions. However, for compatibility with the XMP specification and future extensibility, the list will be kept open so that absolute coordinates could be added later-on.