DIG35 Specification

– Metadata for Digital Images –

Version 1.0

August 30, 2000

Digital Imaging Group, Inc.

The information in this document is believed to be accurate as of the date of publication.

THIS DOCUMENT IS PROVIDED "AS IS." THE DIGITAL IMAGING GROUP MAKES 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 THIS DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

THE DIGITAL IMAGING GROUP WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

About the Digital Imaging Group

The Digital Imaging Group (DIG) is a not-for-profit imaging industry consortium of manufacturers, developers, and users of digital imaging solutions. Originally founded in 1997 by Adobe, Canon, Eastman Kodak, Fuji, Hewlett-Packard, IBM, Intel, Live Picture and Microsoft, the organization has now grown to over 70 members of membership as of August 28, 2000. The primary goal of the Digital Imaging Group is to provide an open industry forum for the exploration, implementation and stewardship of technologies and methods for expanding the Digital Imaging market.

For more information:

Digital Imaging Group, Inc

http://www.digitalimaging.org

Comments should be sent to:

dig35comments@digitalimaging.org

Members of the DIG35 Initiative Group are (in alphabetical order):

Adobe Systems Inc., AGFA-GEVAERT N.V., Canon Inc., Eastman Kodak Company, Fuji Photo Film Co., Ltd., Hewlett Packard Company, Microsoft Corporation, NETIMAGE, PhotoChannel Networks Inc., PhotoWorks, Inc., PictureIQ Corporation, Polaroid Corporation, WorkStation.com

Preface

From consumer to professional and commercial applications, digital images are overtaking the traditional images created with film or video in terms of number of images created and they are also becoming increasingly closer in quality of image. In addition, the increasing sales of digital image capture devices and associated services available for digital image processing will soon result in the proliferation of huge numbers of digital images in many forms and in a variety of sizes and [perceived and measured] value.

From the very beginning of photography (and, before that, in art of all kinds) those who take pictures have tried to capture as much information about the photograph as possible. This information is termed *metadata*.[i]

Every family has shoeboxes filled with photographs, some with a few cryptic notes on the back (which is metadata). Another method of tracking metadata has been in photo albums dedicated to special occasions. In both cases, the information has been limited and often the meaning of the notes is lost over time.

Professional applications make use of digital images as an add-on to their specific skills. Quite often, images are even essential to complete their main job. For example, professional photographers, real estate agents, and insurance adjusters use images as a significant part of required tools for completing the job successfully.

Commercialization of imaging is reaching the highest level of accessibility worldwide. Commercial applications are those where trading of contents is the basis of the business. This would include, for example, clip art vendors, photographic prints, and other image and photography product sales.

Professional and commercial applications have needs to classify, store, archive, transfer, distribute, retrieve, filter, exploit and sell the images and to specifically identify an image similar to the way that families need to identify their pictures. For these applications in today's environment, this is often accomplished by automatic processing software and hardware.

Overall, creating metadata for images enhances our business opportunities, amplifies our memories and increases the joy of sharing. As we share images across distance and time, it becomes even more important to remember as much as possible about the image.

In fact, in today's digital age, sharing images is easier than ever. The growth in digital technology has created a desire among users to manage and exchange their images in a variety of ways including storage, e-mail exchange, Internet/WWW postings and other displays (such as personal electronic photo albums or frames), and printing to a variety of devices with different resolutions.

Indeed, personal processing of images can be organized into four primary areas of use: creation, editing, printing, and sharing. In terms that are more technical, this workflow[ii] is termed input, manipulation, output, and communication. The benefits for digital image use increases significantly by having more extensive metadata that is consistently available throughout this workflow.

First, with digital metadata, the image becomes important not just for today but also in the future. Knowing exactly who, what, when, where, and how a photograph was taken provides a solid basis for documenting our lives and is an integral part of business.

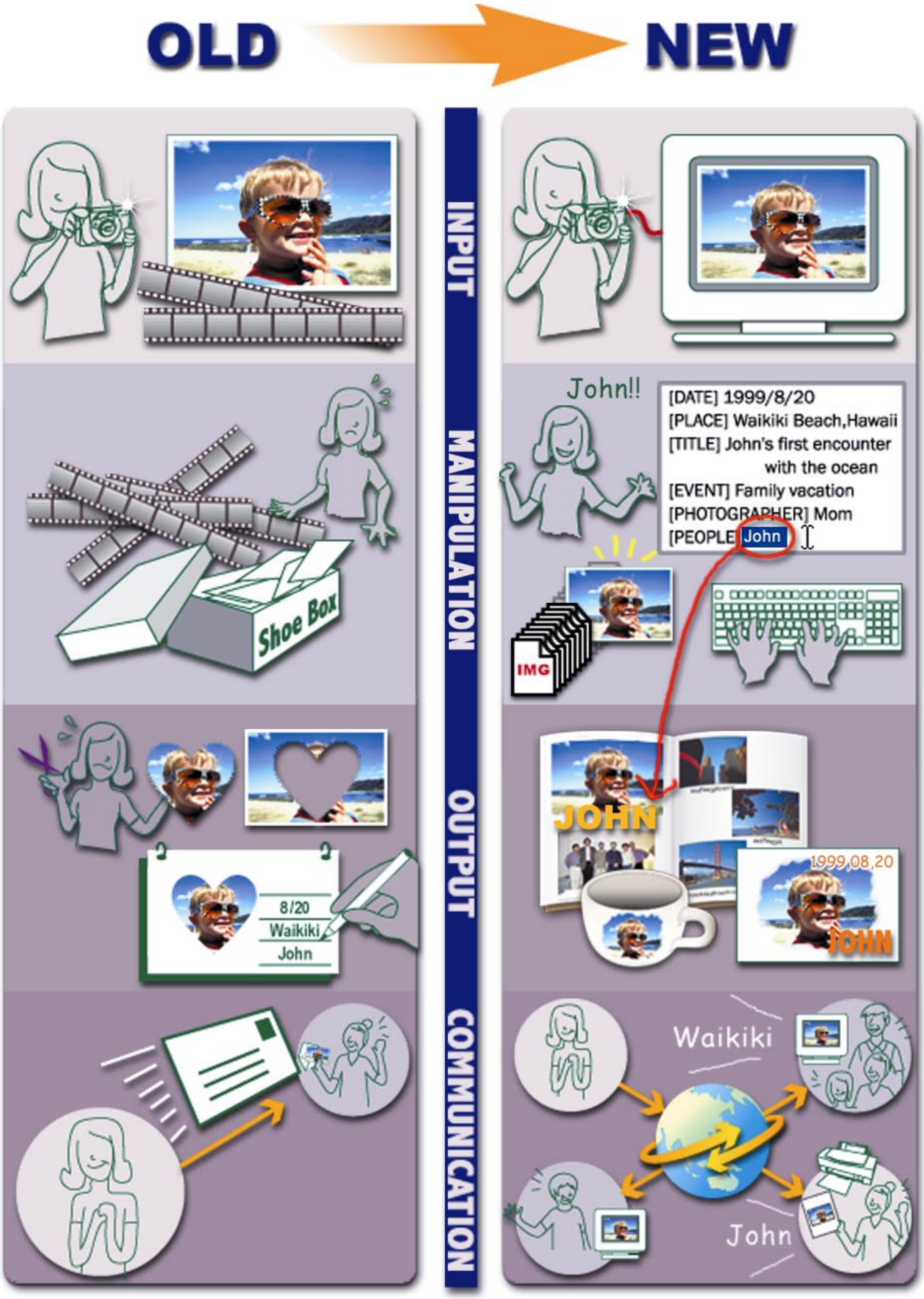
Second, using today's technology, images can be shared far more broadly and in the way most appropriate to each individual and functional requirement. This extensive sharing of images requires more knowledge about the image, especially in a business situation.

Finally, knowing as much as possible about the image enhances sharing. We no longer need to rely on memory alone to provide the background for our most precious history or our corporate assets.

The illustration below highlights several of the fundamental enhancements in today's technology in each of these workflow areas.

[i] The term metadata is composed from the Greek word "meta" meaning aside and "data" meaning the essential content of an electronic container (for example, an image file).

[ii] Workflow defines the progress of work through a series of steps done by an individual, a business, or several different businesses.



[DATE] 1999/8/20
[PLACE] Waikiki Beach,Hawaii
[TITLE] John's first encounter with the ocean
[EVENT] Family vacation
[PHOTOGRAPHER] Mom
[PEOPLE] John

Digital images, whether created by digital cameras or by digitizing video or film-based media, paves the way for many more of these associated services than would be available solely within analog media such as traditional film or VHS video. All of the new services becoming more broadly available will be enhanced by the addition of metadata.

The Digital Imaging Group feels that such digital metadata must be consistently associated with the image — regardless of the computer applications involved in the workflow. To achieve this goal of "persistent" metadata the DIG has created the DIG35 Metadata Specification.

About This Document

The DIG35 Metadata Specification includes the following:

- An overview of the specification, useful for anyone interested in understanding metadata and the potential uses as well as the basics of the DIG35 Metadata Specification. This overview comprises Chapters 1 through 5 of this document.
- A series of technical annexes (Annex A through H) which provides the technical details of the Specification, the XML Reference Implementation, and other information of interest to those who are implementing the DIG35 Metadata Specification in their own work.
- Several Appendices, which provide additional information, background information on metadata in general or the DIG35 Metadata Specification. These Appendices will be useful primarily to those implementing DIG35 into their own work.
- A Glossary of terms and their definitions and references.

Typographical Conventions

Within this document the following typographical conventions are used:

Syntax, and code examples are shown in `fixed width` font.

Italic strings are used for emphasis and to identify words representing variable entities in the text.

Bold strings are used to identify the first instance of a word requiring definition; see each metadata definition or the Glossary for details.

Links are shown in this color font.

NOTE: Text that is of special interest is represented as a note to the reader.

```
Schema Definitions
```

```
Examples are shown in these boxes.
```

Terminology

Within this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in RFC 2119. However, for readability purposes, these words do not appear in all uppercase letters in this specification.

Status of This Document

This document is a recommendation of the DIG35 Initiative Group. Comments on this document should be sent to <`dig35commnets@digitalimaging.org`>.

Contents

List of Figures

List of Tables

Chapter 1: Introduction

Metadata is additional information that is associated with the primary data. In the context of this specification, it is “additional data linked with the image data beyond the pixels which define the image.” Metadata, to be most valuable for the owner(s) and user(s) of an image, needs to be consistently maintained throughout the image lifecycle. In today's environment of image editing applications, rapid transmission via the Internet, and high quality photographic digital cameras and printers, the lifecycle of a digital image may be very long as well as complex.

However, if standards for managing and exchanging the metadata for these images, including associated data input methods for users, do not materialize, we will be left with the digital equivalent of the "shoebox" and will not realize the full potential of digital imaging power and flexibility.

Realizing this the Digital Imaging Group formed an Initiative Group in early 1999 to begin addressing the issues involved and provide the first step towards a standard set of metadata for digital images. This Initiative Group (commonly called DIG35) includes representatives from Adobe Systems Inc., AGFA-GEVAERT N.V., Canon Inc., Eastman Kodak Company, Fuji Photo Film Co., Ltd., Hewlett Packard Company, Microsoft Corporation, NETIMAGE, PhotoChannel Networks Inc., PhotoWorks, Inc., PictureIQ Corporation, Polaroid Corporation, and WorkStation.com.

The result of this broad collaboration of imaging and technology industry participants is this technical specification defining specific metadata for digital images and a recommended implementation model. In total this set of documents comprises the "DIG35 Metadata Specification for Digital Image Metadata" or DIG35 Specification (often referred to simply as "DIG35").

1.1 Vision and Goals

The vision of the DIG35 is:

> “To provide a mechanism to allow end-users to see digital image use as being equally as easy, as convenient, and as flexible as the traditional photographic methods while allowing benefits that are possible only with a digital format.”

The overall goal of the DIG35 initiative is to define a standard set of metadata for digital images that will improve the semantic interoperability between devices, services and software. This interoperability will make it easier to organize, print, exchange and otherwise process digital images in this new millennium.

1.2 Scope

This specification defines a set of ***public metadata*** for digital still images. These images may encompass a single image or a collection of images that could be supported and exchanged by current and future devices, software, and services in an open environment. In addition to the metadata definition, a recommendation for implementation is defined to enable exchange of such metadata.

This specification does not concern itself with vendor-specific proprietary ***private metadata*** or a particular application domain. Private or application domain-specific metadata should be defined within their respective organizations and, as such is out of the scope of this specification.

Additionally, the DIG35 Specification does not mandate procedural metadata[i] such as processing parameters which changes the visual appearance *after* decoding the image data (e.g. cropping, rotating etc.), printing information (e.g. number of copies, output size etc.) and ordering information (e.g. delivery and payment information). There are, however, exceptions at this time. Print Aspect Ratio (PAR) metadata is an example that does not follow this rule. Technical metadata that are well-defined and intended to produce better quality images that benefit the users without additional user effort are considered exceptions.

[i] Such metadata are either discussed in other organizations or subject to future discussions and efforts by the Digital Imaging Group.

1.3 Design Requirements

The design goals and principles of this specification are identified below to give a high-level direction of this specification.

1.3.1 Design Goals

The design goals of the DIG35 initiative are to define a metadata set that is:

- INTERCHANGEABLE: DIG35 is based on a sound conceptual model that is both generally applicable to many applications and assured to be consistent over time. DIG35 will create a better environment for locating and (re) using specific images.
- EXTENSIBLE AND SCALEABLE: DIG35 enables application developers and hardware manufacturers to add additional metadata fields. This allows future needs for metadata to be fulfilled with limited disruption of current solutions.
- IMAGE FILE FORMAT INDEPENDENT: DIG35 does not rely on any specific file format and can therefore be supported by many current and future file formats and compression mechanisms.
- CONSISTENT: DIG35 can work well with existing standards today allowing the metadata standard to be widely acceptable and usable in a variety of application domains and user situations.
- INTERNET-READY: DIG35 provides seamless integration with the Internet by utilizing XML (the recommended implementation method).

1.3.2 Design Principles

Design principles adhered to when developing the DIG35 initiative were developed prior to beginning the technical efforts. These principles state that the DIG35 initiative shall:

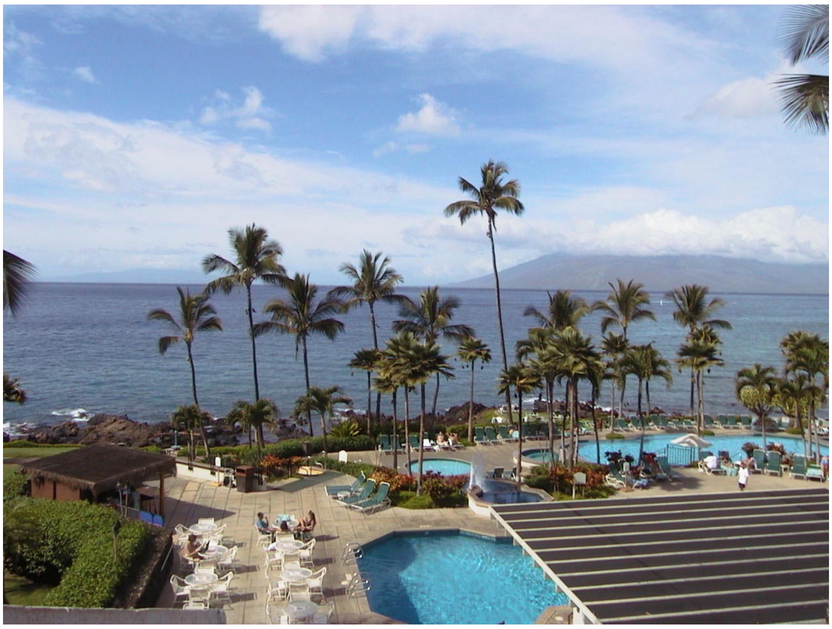
- use existing standards and output of other organizations as much as possible while creating a future-looking metadata standard.
- focus on mid and long-term perspectives, not only on what current digital imaging technology may be able to offer today
- be simple for developers to utilize but sophisticated enough to cover a wide spectrum of features
- support *information* preservation, not *data* preservation
  Many devices, such as digital cameras, may store metadata in a format that end users are not familiar with which will discourage use. Thus, applications may need to apply appropriate conversions to transform these values into user-understandable formats.
- allow metadata redundancy
  Certain values exist that can be calculated from other fields, at the definition level, redundant metadata do exist and need to be managed appropriately. Where this is applicable the specification highlights the fact and makes recommendations as to how to manage such redundancy.

Chapter 2: Metadata Architecture

2.1 What is Metadata?

As previously discussed, metadata is additional data linked to the image data which provides additional information about the contents of the image, the creation of the image, or the uses of the image. Metadata may be used in a variety of ways, including:

- Providing in-depth information on the image and its creation, such as date, time of day, focus distance, light levels, use of flash, GPS location, etc.
- Allowing easy indexing, identification, categorization and usage-control according to any pre-determined schema, such as image type, copyright conditions, originator, subject matter, location, etc.
- Enhancing the intrinsic content of the image, such as differentiating between several different beach images, which may look the same to an unknowing observer.



Example metadata:

Photographer: Kats Ishii

Date: 1999/12/09

Location: Maui, Hawaii

Title: Memorable conference

Comment: View from restaurant

Event: Committee meeting

Figure 2-1: Sample image with metadata

2.2 Image Model

Digital images are commonly stored in well-defined file formats, e.g. JFIF (JPEG), Exif[22], TIFF[1], Flashpix[3], or others. Although there are differences in how the data is physically stored on the recording media, conceptually each format has a similar Image Model. The illustration below shows the logical structure of the basic image model in use today. This is used as a reference model throughout the rest of this document and for DIG35.

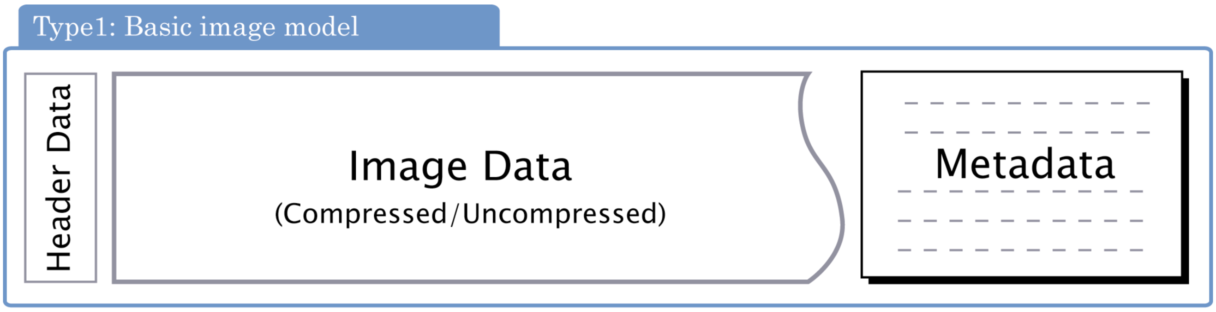


Figure 2-2: Basic conceptual structure of a digital image

The “Header Data” is a specific set of metadata that consists of essential information to decode the image data (such as the image size, bit depth, colorspace and compression method, etc.) plus additional information required to locate the image data and the other metadata within the file. While this information is metadata of the image data by definition, it is tightly coupled with the image data and specific to a particular image file format. Thus, the detail definition of the "Header Data" is considered a mandatory entity of the image file format. As previously noted, the definition of file format is not the scope of this specification.

The “Image Data” is the primary data of a digital image[i]. It is the digitally encoded representation of a picture that is usually compressed but may also be stored as uncompressed raw signal data of the capture device. Any other related non-image data, either directly or indirectly, are considered as “metadata” in this specification.

Metadata may include information such as the capture conditions of the camera, a description of the image content, and intellectual property information. The method of storing metadata within an image file format varies from format to format. Variants of a bitmap format, for example, do not define placeholders for metadata. JFIF (JPEG), on the other hand, defines a mechanism to allow metadata to be associated with the image data. However JFIF (JPEG) does not currently define the type of metadata to be stored. This means that the storage mechanism and metadata structure must be defined by each application in order to exchange metadata associated with that file. Lastly, Exif, TIFF, SPIFF and Flashpix are examples of file formats that define a rich set of metadata along with a storage mechanism. Defined metadata in these file formats includes descriptional information such as the title, date/time of the capture, the capture source (e.g. camera, scanner) from which the image was created, and copyright information. There is compatibility of these file format based metadata definitions to some extent; however, not all metadata can be mapped from one format to another. This lack of direct mapping causes a variety of problems when conducting a simple image file format conversion.

Figure 2-3 shows additional models that represent other file formats. Note that each file format, while taking a unique approach, consists of the same three major components: the header information, the image data, and metadata.

[i] Other file format specifications may term it as either codestream or bitstream.

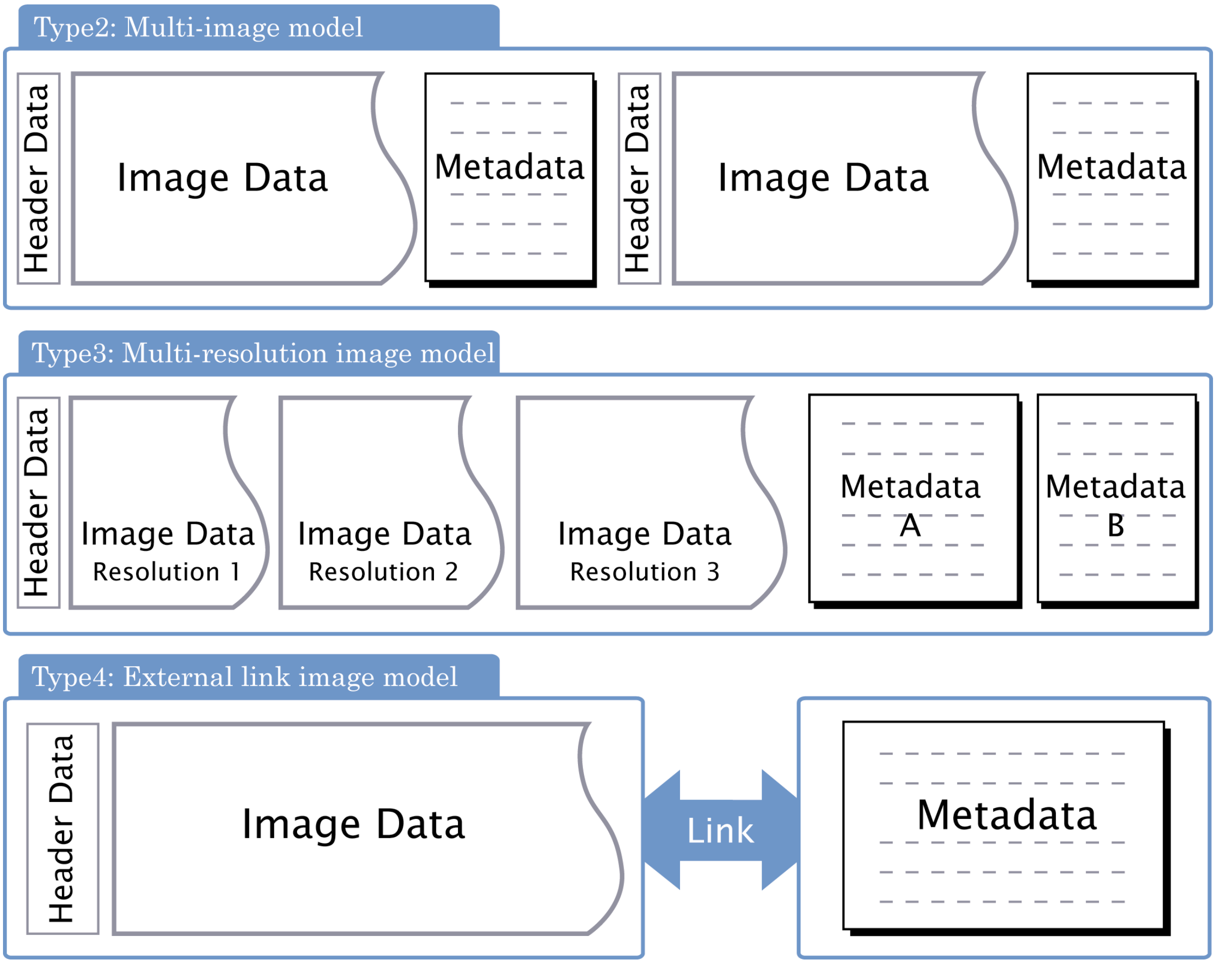


Figure 2-3: Additional conceptual structure of a digital image

In addition to the metadata that is closely associated with the file format image data, there are other metadata that are used throughout the lifecycle of the image data. Such metadata may include print condition information such as output media size, the number of copies and service order information (i.e., delivery instructions), and payment methods. While this information may be very useful in various applications, it is out of scope of this specification.

2.3 Metadata Interchange Model

2.3.1 Image Pixel Data Interchange

The most important aspect of digital imaging is to be able to view, share, and enjoy the actual image in a variety of ways. Those images are, in most cases, captured directly with a digital camera or created by scanning a film or prints. These digital images are stored as an image file on a storage media to be used in various ways to fulfill the objectives of the user. Typical uses, for example, are opening an image into an application to be printed for personal pleasure or used to create post cards and/or web pages to share them with family members or friends. The fundamental value to the user is the capability of viewing the captured image in a variety of ways and at different times.

This type of image data interchange and work flow are enabled due to the fact that current devices and applications support an image file format model which is well defined and standardized. Thus, the actual image data can be interchanged between various file formats.

Figure 2-4 shows a typical model of image data interchange. Filters include image libraries or plug-in type applications that enable the system to read and store the image data into its native (proprietary) data format. As long as both systems support the file format model, the image data can be successfully interchanged.

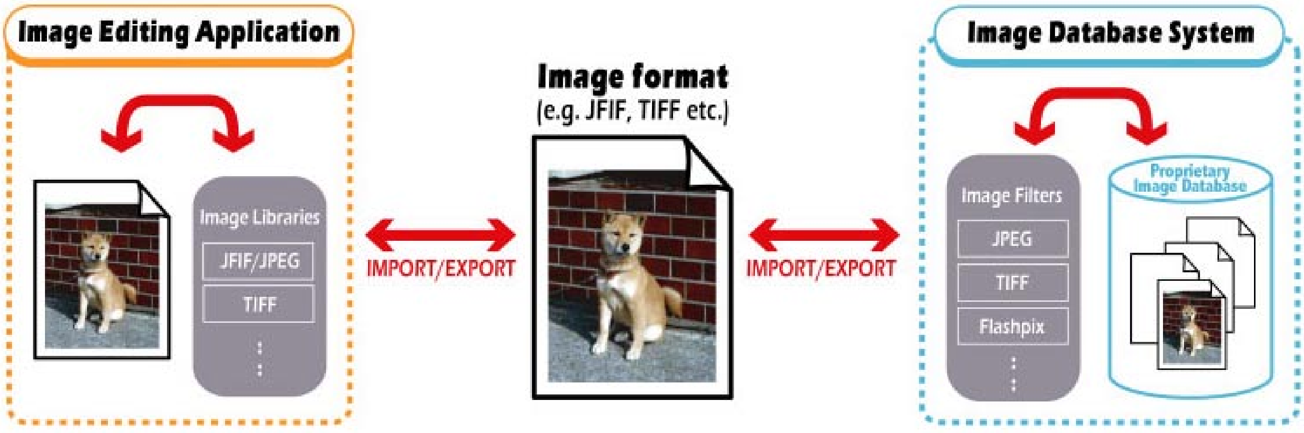


Figure 2-4: Image data interchange model

2.3.2 Image Pixel and Metadata Interchange

While it is true that devices and applications support image data interchange, the metadata that are to be associated with those images are not always passed on within the imaging workflow. In application terms, the persistence of metadata is not implemented. There are many reasons for this lack of support by applications. Among the most important ones are:

- The file format does not support metadata that the application uses.
- The metadata is stored in a proprietary file format that is not publicly defined.
- A standard image metadata definition and interchange format model does not exist.

Therefore, users are not able to reuse the metadata that have been automatically recorded by the capture devices or the information that they may have entered. With a standardized file format-independent metadata definition and interchange format, users will be assured that those metadata previously entered are properly retained and treated as the important information that it represents. Users will thereby be encouraged to input more metadata and gain more benefit from the superior technology digital imaging offers.

Thus, this specification consists of two major parts: Part 1, the metadata definition and Part 2, the reference implementation for interchange. The specification in its entirety is designed based on the following principles, which encompass vendor-neutrality and technology-independence.

- Partition the metadata definition into logical blocks; devices with limited resources may choose to support portions of this definition and those applications that need further detail can easily extend it’s scope
- Use a widely accepted, flexible and extensible implementation format (XML)

The metadata interchange model is shown in Figure 2-5. Similar to that of the image data interchange model, metadata is also imported/exported via a standard metadata reader/writer ability that can map the information into its system’s database/data structure. This specification includes one such implementation; see Annex G for information on the DIG35 reference encoding method — XML. It is ideal for imaging tools to share a common metadata model within each application or system that will enable easy sharing of metadata from one application to another. However, given the diversity of digital image devices today and in the foreseeable future as well as applications in a wide range of spectrum, it is neither feasible nor realistic to standardize a single unified internal metadata model that is sufficient for all purposes.

NOTE: There are additional challenges attributed to associating metadata with images and the problems that are encountered relating to retention of image metadata throughout the lifecycle of that image. See Appendix II for detail discussions.

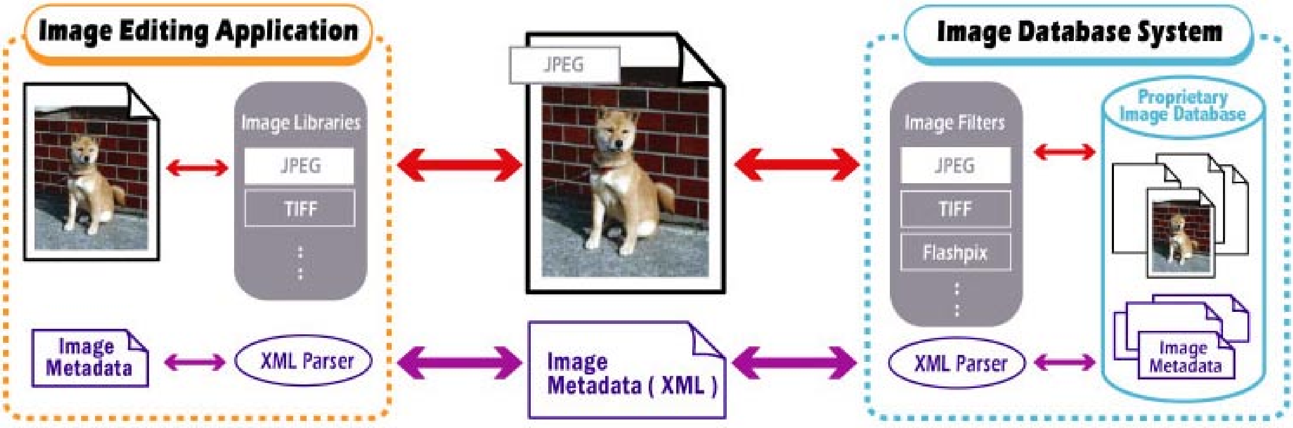


Figure 2-5: Metadata interchange model

2.3.3 Challenges of Metadata Interchange

Vendors and users alike recognize the value of image metadata. However, the rapid proliferation of data manipulation and management tools has resulted in a wide variety of information technology products that process metadata differently, or not at all, and without much consideration for sharing or exchange of metadata. Thus the challenge is to define a standard set of metadata that is generally acceptable by vendors, users and content providers. The challenges include:

- creating a well defined semantic model that promotes interoperability of metadata
- creating proper rules and guidelines for retention and association of metadata with image data
- educating users and providing guidelines

2.4 Metadata Storage Model

A special case of image interchange is image storage. For image storage, the metadata could be (and perhaps typically is) separated from the image data. The interchange file would be “burst” and the metadata might populate a relational database. This raises several interesting challenges for the management of images, but also offers significant advantages.

Once burst, the image management system can optimize how it manages the image and metadata individually. The most obvious example of this is the difference in physical size of the data. Image data might be stored on a staged archival access mechanism (disk → tape), while the metadata might be stored on magnetic. Another example would be to optimize around how the data is accessed and used. If the metadata is only used after an image is accessed, it could reside on staged access media along with the image. If the metadata is used for image search, it could be loaded into a Relational or an Object Oriented database. In some cases, a mixture of these approaches could be used.

However, once burst, the image management systems has lost the physical relationship between the image and metadata. If it becomes necessary to access an exact replica of what was received, this may be difficult. Whether this is important or not is yet to be determined in the marketplace. However, there is significant effort being made to protect images using watermarks and encryption technology. File formats are beginning to make use of Intellectual Property Rights (IPR) properties (“tags”) to indicate who owns or created images. If these properties apply to the collective image and metadata it becomes a requirement to track the items as a group. This could be difficult once a file is burst.

2.5 Progressive Metadata Model

Most closely aligned with the capabilities of progressive compression and file formats such as JPEG 2000, the progressive metadata model allows a client to access metadata only when as the client needs the additional information. This is an extension of the storage metadata case that allows the user to access the metadata that is needed for a particular purpose in succession and only as needed. As in the “burst” storage case, the integrity of the original image has been lost, and a particular user may never have the complete image metadata set and may not be aware of this fact.

Chapter 3: DIG35 Image Metadata Definition

3.1 Overview

Image metadata is a "building block" for digital imaging that may be used within the wide spectrum of the imaging workflow. This specification defines a standard set of image metadata based on a generic concept that may be further divided into conceptual "blocks." Each of these blocks describes a unique aspect of the image. By partitioning metadata into discrete blocks, developers may extend a particular block without affecting the entire architecture thereby ensuring semantic interoperability while allowing vendors to add value to the metadata and image data itself. The DIG35 metadata blocks are shown in Figure 3-1.

Note that procedural metadata are out of scope of this specification. Examples of procedural metadata include:

- Parameters for image processing that change the visual appearance by arbitrary cropping, rotation, or other transformations.
- Order information to specify product and quantity.
- E-commerce information such as billing or payment data or delivery addresses.

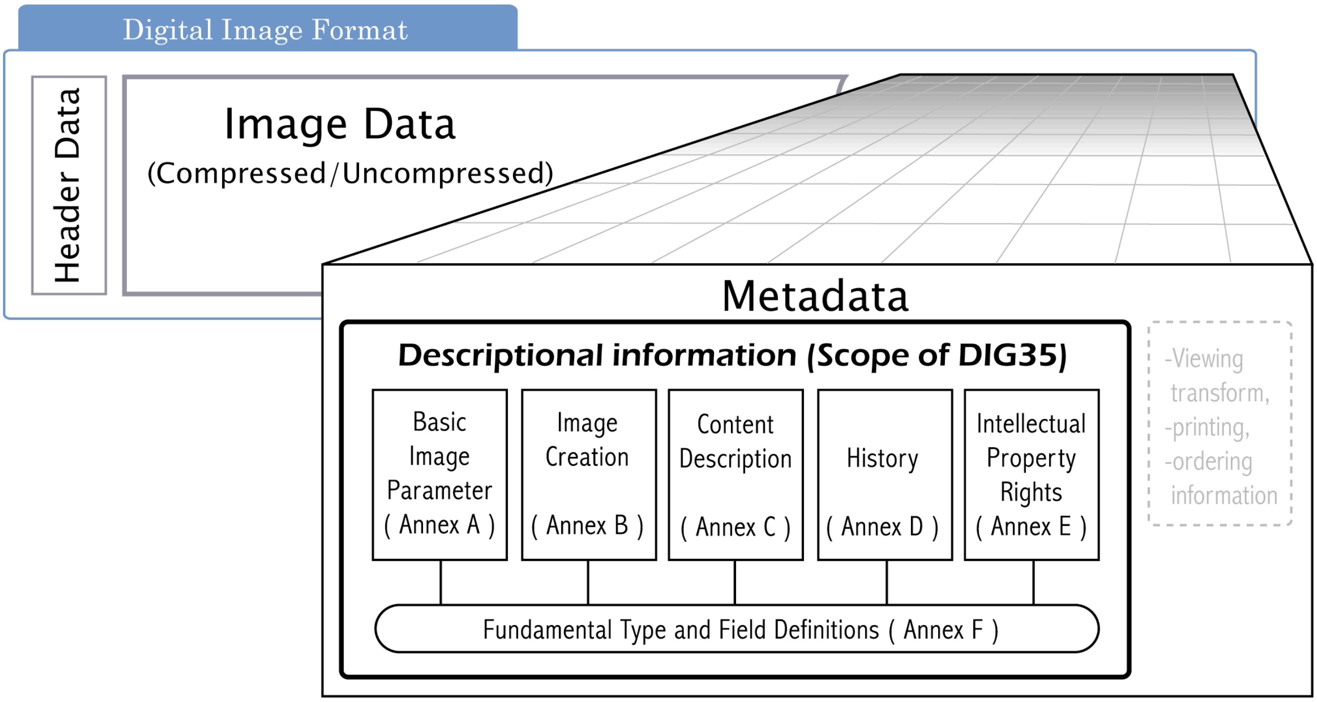

Digital Image Format
Header Data
Image Data
(Compressed/Uncompressed)
Metadata
Descriptional information (Scope of DIG35)
Basic Image Parameter (Annex A)
Image Creation (Annex B)
Content Description (Annex C)
History (Annex D)
Intellectual Property Rights (Annex E)
Fundamental Type and Field Definitions (Annex F)
-Viewing transform,
-printing,
-ordering information

Figure 3-1: Scope of this specification within the Image Model

3.2 DIG35 Metadata Blocks

DIG35's Metadata definition consists of five logical blocks with a separate common definition that is referred to by other blocks. While each block is logically partitioned, they may be linked to each other to form additional semantics.

3.2.1 Basic Image Parameter Metadata

Since this specification is a general-purpose metadata standard, it must be applicable to the broadest possible class of file formats. Since each file format makes distinct decisions regarding what elements are important as header information, it is impossible to delegate the specification of header metadata to file format developers. In fact, this specification takes the opposite approach and assumes the existence of a file format that contains no header information. This assumption ensures that any format may be transcoded into another file-format.

In order to do this, a block of metadata is defined that contains information similar to, and identical in use to, file header metadata. There should never be conflicts between this block and the file header metadata as this block is intended to be used, as stated above, *only* when there is no file header metadata. However, if there is a conflict between the file-format header information and the Basic Image Parameter Metadata, the file header should always take precedence. See Annex A:Basic Image Parameter Metadata for details.

NOTE: Each file format defines its own header information; this is very much dependent on the features the format supports. Those features are specific to the image data and thus out of scope of this specification. The DIG35 Basic Image Parameter metadata should be considered *informational* and not be used to decode the image data stored in the associated file.

3.2.2 Image Creation Metadata

The Image Creation Metadata defines the "how" metadata that specifies the source of which the image was created. For example, the camera and lens information and capture condition are useful technical information for professional and serious amateur photographers as well as advanced imaging applications. See Annex B:Image Creation Metadata for details.

3.2.3 Content Description Metadata

The Content Description Metadata defines the descriptive information of "who", "what", "when" and "where" aspect of the image. Often this metadata takes the form of extensive words, phrases, or sentences to describe a particular event or location that the image illustrates. Typically, this metadata consists of text that the user enters, either when the images are taken or scanned or later in the process during manipulation or use of the images. See Annex C:Content Description Metadata for details.

3.2.4 History Metadata

The History Metadata is used to provide *partial information* about *how* the image got to the present state. For example, history may include certain processing steps that have been applied to an image. Another example of a history would be the image creation events including digital capture, exposure of negative or reversal films, creation of prints, transmissive scans of negatives or positive film, or reflective scans of prints. All of this metadata is important for some applications. To permit flexibility in construction of the image history metadata, two alternate representations of the history are permitted. In the first, the history metadata is embedded in the image metadata. In the second, the previous versions of the image, represented as a URL/URI, are included in the history metadata as pointers to the location of the actual history. The history metadata for a composite image (i.e., created from two or more previous images) may also be represented through a hierarchical metadata structure. While this specification does not define the "how" or "how much" part of the processing aspect, it does enable logging of certain processing steps by adding textual descriptions of operations applied to an image as hints for future use. See Annex D:History Metadata for details.

NOTE: Neither the processing nor the history metadata specify exact values of the actual processing steps or the compositing steps. Such metadata is reserved for further study.

3.2.5 Intellectual Property Rights (IPR) Metadata

The Intellectual Property Rights Metadata (IPR) defines metadata to either protect the rights of the owner of the image or provide further information to request permission to use it. It is important for developers and users to understand the implications of intellectual property and copyright information on digital images to properly protect the rights of the owner of the image data. See Section 5.6 for further discussions of intellectual property rights issues. See Annex E:Intellectual Property Rights Metadata for details.

NOTE: Several international organizations are looking at standards in the area of intellectual property rights.

3.2.6 Fundamental Metadata Types and Fields

The Fundamental Metadata Types define the format of the field defined in all metadata blocks. Those may include a collection of metadata fields such as an “address type” or a representation of an attribute of other fields such as the “language attribute.” The Fundamental Metadata Fields define fields used in all metadata blocks. These include a definition for language specification and timestamps. Annex F:Fundamental Metadata Types and Fields Fundamental Metadata Types and Fields for details.

3.3 Example Workflow

The work flow of digital imaging varies from that of the analog image processing and work flow. Some of the differences in the two work flows are highlighted in the examples below. The DIG35 use-cases provide additional insight into the digital image work flow issues that DIG35 are concerned with in this specification.

NOTE: See companion document “DIG35 Requirements and Use-cases” for further detail of DIG35 use-cases.

3.3.1 Digital Image Process

A typical imaging work flow including metadata retention consists of three basic processes; “Creation”, “Management/Manipulation” and “Output.” An example work flow is illustrated in Figure 3-2 that shows the type of metadata that could be added to the image data. By persisting the metadata recorded in the previous process, the next application or device would be able to take advantage of those information and use them appropriately without involving duplicate user inteactions. The dotted line denotes the metadata that are in scope of this specification.

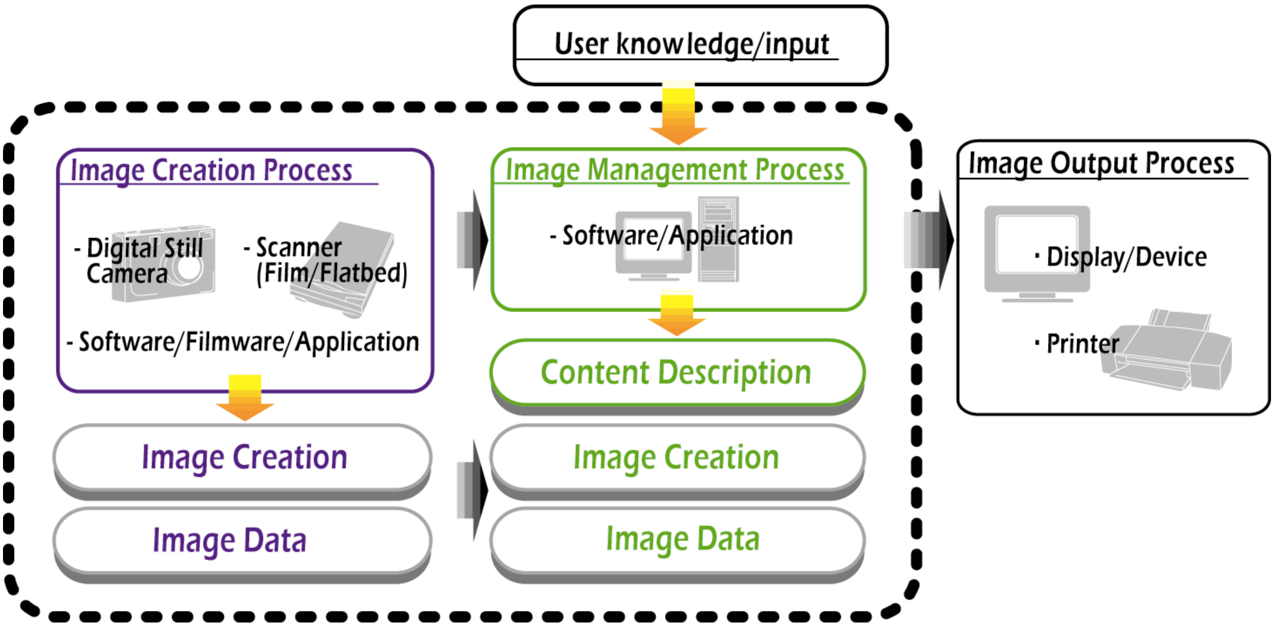


Figure 3-2: Example image work flow by process

3.3.2 Interactive Metadata Access/Usage

The following figure (Figure 3-3) shows interoperability of several applications that all process the same digital image with metadata.

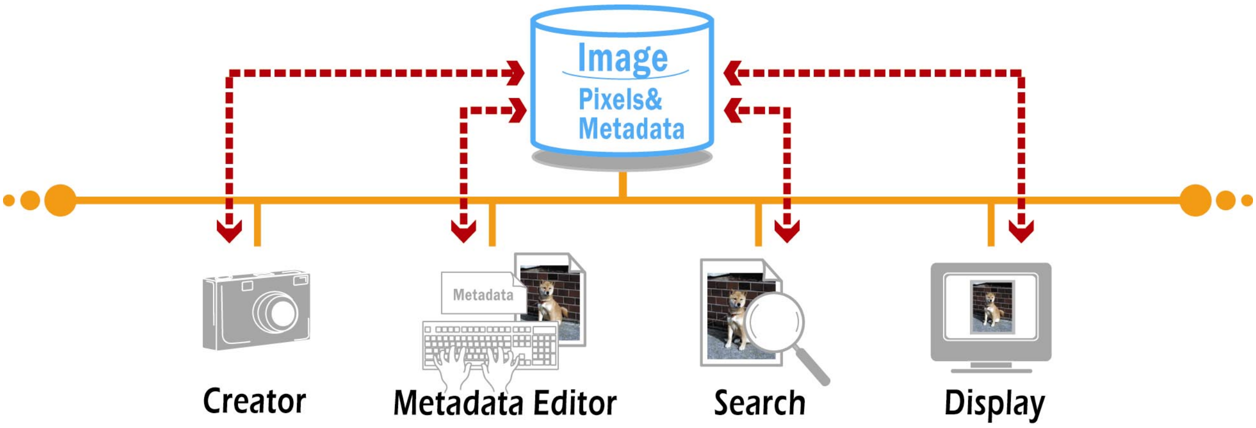


Figure 3-3: Different metadata access/usage example

The heavy dashed lines indicate information flows accessing the metadata. The same metadata must work with all of the applications, for example creation, editing, search, and display

The disk icon represents an image file, with metadata. The applications represented are typical, but certainly not limited to the particular types of applications shown. The heavy dashed lines represent the information flows that allow the applications to get and set the metadata. In this example, the image file resides in a single computer. The applications may run on the same computer, or may run on different computers with networked drives. This means that the information flows are not protocol-based interchanges. Instead, they represent application programming interfaces allowing the programs to access the metadata. It is assumed that the applications run one at a time, so there is no need to for record locking and associated complications.

The applications each read and parse the XML data. This builds a model of the metadata in the application memory. The application may just read the metadata, and not modify it. Alternatively, an application may update or delete existing metadata, and may create additional metadata. Periodically (or else just before exiting) the application updates the image's metadata. To do so, it creates the XML from the internal metadata model stored in its memory, and then writes the XML to the file.

This promotes interoperability of the same data file among and between several independent applications, all based on the common information model and format of the DIG35 metadata.

Chapter 4: DIG35 Implementation Definitions

4.1 Metadata Encoding

This specification recommends using the World Wide Web Consortium (W3C) XML (Extensible Markup Language) [28] as the standard metadata interchange format. While other implementation methods are certainly possible, the primary reasons for choosing XML include:

- XML is already widely adopted as a cross-platform and Internet-enabled implementation language;
- XML supports multiple languages (regardless of the character set) in a single document;
- Many applications within the imaging workflow can interface to XML structures;
- XML provides a highly extensible method for creating device-independent, language-independent, and application-independent interchange formats;
- XML has built-in multi-language support to support different languages can be used in a single document;
- XML is equally well-suited for handling relational or hierarchical data structures;
- XML provides a solid foundation for implementing both human- and machine- readable and understandable metadata.

The industry has already developed a robust set of tools for both writing and reading XML. It offers a ready-made environment in which developers of digital imaging applications and devices can rapidly incorporate the advantages of metadata structures. In addition, the widespread usage of XML within the Internet community provides for an inherently smooth meshing of the metadata structures with underlying network transport mechanisms and Web-based communication methods.

The syntax to group all metadata sub-blocks (defined in Annex A though E) into a single XML document is defined in Annex G:DIG35 XML Document Definition.

4.1.1 Schema Definitions

There are two aspects of defining a metadata field: describing its semantics and the syntax. The semantics of the metadata is provided with a textual description for each field. Metadata fields that construct a logical meaning are grouped together where the syntax of individual fields is defined within that group.

The syntax is described using W3C's XML Schema language[31][32]. The XML Schema language is used as it clearly defines the type of each metadata field, and the relationship between fields.

XML is by no means the only encoding method for DIG35 metadata. The types of the fields and the relationship between the fields, described using XML Schema throughout this document, could also be used to describe DIG35 metadata stored, for example, in a relational database or a binary image file.

NOTE: W3C's XML Schema is *not* a W3C Recommendation at the time of the publication of this specification, thus the actual syntax defined here is subject to change. However, the datatypes and relationships between fields that are represented as a schema should be consistent. The introductory document XML Schema Primer[30] provides details on the XML Schema usage used in this specification. DIG35 supports the W3C's XML Schema effort and intends to revise the DIG35 recommended XML encoding method following publication of the XML Schema Recommendation by W3C.

4.1.2 Namespaces

XML namespace is a collection of names, identified by a Universal Resource Identifier (URI). This allows XML documents from different sources to use elements with the same names, and then be merged within a single document without confusion. Since DIG35 metadata may either incorporate other metadata for extensibility or be used in other applications, it is important to define an XML namespace specifically for DIG35 elements and attributes.

Three namespace abbreviations are used throughout this specification. These namespaces are for XML, XML Schema and DIG35 definitions.

The XML namespace:

```
    xmlns:xml="http://www.w3.org/XML/1998/namespace"
```

The XML Schema namespace:

```
    xmlns:xsd="http://www.w3.org/1999/XMLSchema"
```

To specify the DIG35 XML namespace the following URI is defined. It should be used by experimental implementations:

```
   xmlns:dig35="http://www.digitalimaging.org/dig35/1.0/xml"
```

4.1.3 Syntax Conventions

In this specification, several syntax conventions are used. Table 4-1 summarizes the syntax conventions used in the DIG35 recommended XML implementation. Additionally, examples of encoding of the metadata are described in XML. Note that for readability purposes, most examples are well formed XML focused on fields that are related to the metadata definition.

Table 4-1: Syntax conventions

	Description	Example
DIG35 defined types	DIG35 defined type name starts with “t” prefix. Following the prefix, subsequent words shall begin with an upper-case letter. Multiple words are combined without spaces in between.	tDateTime
XML elements and attributes	All names shall use all upper case letters. Subsequent words shall be connected by an underscore ('_').	GENERAL_INFO
Enumerated values	All words begin with upper case letter. Multiple words are combined with a space (‘ ‘).	`Digital Camera`

4.1.4 Extensibility

This specification defines public metadata that is applicable to general images. Thus, its coverage may not be enough for a particular application domain that needs further detail to be stored. DIG35 metadata documents therefore should allow additional elements to be embedded, keeping the integrity of the overall framework, allowing private metadata to be 'plugged-in' that only a particular systems may be able to understand and use such information.

Chapter 5: Miscellaneous Topics

5.1 Association of Metadata and Image Data

There are many challenges attributed to metadata usage within the imaging workflow. After conforming to a standard metadata set, the next most significant challenge is associating the metadata with the image throughout a standard workflow. There are several alternatives — each having benefits and drawbacks that may create new problems to be encountered relating to retention of image metadata throughout the lifecycle of that image. A detailed analysis of metadata association and possible metadata association methods is discussed in Appendix II.

5.2 Redundant Metadata

5.2.1 Redundant Data Within the Metadata

Some of the metadata fields may cause redundant data to be stored. For example, if the date an image taken at a birthday party, the subject's birthdate, and the subject's age are all stored as metadata, then there is the potential that these three fields may conflict.

Calculation of missing redundant data is certainly expected. For example, if a search engine is searching for images of people approximately 40 years old, then calculating their age from the image date and the birthdate is possible. Such interaction of fields is beyond the scope of this specification.

If redundant information is inconsistent, the file should still be treated as valid. There is a possibility that calculated information produces an inconsistent (in error) value. This specification does not define the mechanism for solving this situation. The DIG or others may provide "Best Practices" documents to help developers and users with such situations.

5.2.2 Redundant Data Between Metadata and Image Data

Where metadata duplicates the image data (for example the number of pixels may be stored in the metadata and in the image itself), there is the possibility of redundant and inconsistent data. Where there is redundant data (such as in the basic image parameter block), the metadata should not be used to decode the image. The metadata is useful, especially in a system where the overhead (time and/or complexity) is large and the image is not interpreted, values stored in the metadata can still be used to classify the image. For example: A user may specify to only search for high-resolution images.

NOTE: The decision on whether to recognize the metadata or the image data as the correct data when there is redundancy is still under investigation. DIG35 welcomes comments in this area.

5.3 Inconsistent Metadata

Each of the metadata fields can contain a timestamp. This timestamp can be used to determine whether the metadata is newer or older than the image data. Where the metadata is newer than the image data, then the metadata can be assumed to be consistent.

Where the image data is newer than the metadata, then the metadata may not be consistent with the image, and especially fields like the location need to be checked for accuracy before use.

5.4 Metadata Compression

Many standards organizations and commercial developers have invested significant time and effort to create efficient compression algorithms for digital images. However, compression of metadata has not been explored and developed to the same depth.

In general, high-quality image data is magnitudes greater in size than the metadata associated with it. Yet, when the image size is reduced, for example to represent a thumbnail, the size ratio is reversed. Size of the metadata quickly becomes an issue for both users and developers when this occurs.

DIG35 has identified requirements for metadata compression systems. One of the recognized requirements from the DIG35 use-cases is that a file formats that has the capability to store compressed metadata must store header information that describes the compression algorithm and any necessary decompression parameters. This is to ensure that a program reading the metadata as a byte stream has the minimal information required to make use of the metadata *before* it actually sees the metadata. An additional requirement recognized is that file formats, which store compressed metadata, must compress significant blocks of metadata *en bloc* and not create a large number of small, independently compressed segments.

Several ad-hoc experiments have demonstrated significant compression in moderately sized XML metadata is possible using currently available compression algorithms. Reduction in metadata sizes of 50 – 70 % has been observed using one widely available commercial compression package. The Wireless Access Forum (http://www.wapforum.com) has defined a wireless access protocol (WAP), which includes a component for binary representation of XML. This technique may also yield significant metadata data compression.

Based on these experiments and observations, it is premature to select a particular compression scheme. However, these results are conclusive enough to determine that a recommended practice is necessary rather than any additional compression algorithms defined specifically for metadata. Such definition is beyond the scope of this specification.

NOTE: DIG35 is investigating the potential inclusion of a "null compression" algorithm which would be defined so that all metadata would have the compression algorithm, version, and parameter identification metadata for consistency. DIG35 welcomes comments in this area.

5.5 Security Considerations

The full spectrum of possible security risks associated with image metadata is beyond the scope of this specification. However, this section addresses the importance of authentication of metadata and of verification that the pixel data and metadata are properly associated.

There are a number of security risks associated with metadata. The security mechanisms for authentication and privacy address some of these risks. The tolerance level for risk varies widely, as does the likelihood of a security violation. The typical digital images of consumers are not likely to need security methods as extensive as those of a governmental or commercial image bank. In many consumer instances, the concern is not of overt (or covert) threats to their images and metadata, but rather a concern that there might be accidental changes to metadata, or that an image is incorrectly associated with metadata from a different image.

Some applicable security mechanisms include:

- Hash generation—a technique that creates a hash value from given data. Hash algorithms should have these properties—(1) changes in the given data should result in different hash values, and (2) computation of given data that has a particular hash value should be difficult.
- Encryption—a technique that creates apparently random data from the given plain text. Encryption algorithms should have the property that deriving the original data from the encrypted data should be difficult without the decryption key, and should be easy with the decryption key.
- Signature generation—a technique that combines hashing and encryption to create a signature which indicates an association between a given set of data and a particular entity, which is said to have signed the data. A digital signature technique should have the property that changes in the given text should result in changes in the signature, creation of given text resulting in a particular signature should be difficult, and creation of a particular digital signature without knowing the secret key of the original signer should be difficult.

It is useful from the DIG35 point of view to be able to show:

1. that metadata has not been altered after some given time or version;
2. that the image data has not been altered after some given time or version;
3. that some metadata is properly associated with some image data.

Hashes or digital signatures of the metadata or image data can be used to show that the image pixel data or metadata have not been changed (1 and 2 above). A hash or digital signature of the combined (concatenated) hashes or signatures allows verification that the proper association between metadata and image pixel data. The technical specifications for these mechanisms are for further study.

Privacy of image pixel data or metadata is out of scope for DIG35.

5.6 Metadata Persistence

Rules for metadata persistence are a fundamental component of a metadata systems design. If persistence were merely an issue of copying all metadata, then the implementation of a mechanism for metadata persistence is a manageable task.

It is not the case, however, that blindly copying metadata suffices. The complexity of metadata persistence is related to the set of reasonable expectations about the quality of the metadata following persistence. If the image is cropped, it is reasonable to expect that the “Content Description” metadata be updated so that it only describes objects relating to the image after cropping.

The following are a set of desirable properties for metadata. They are listed in increasing order of importance, and increasing difficulty of implementation

- Metadata should be well-formed, as defined by XML (or other encoding rules).
- Metadata should be structurally valid, as defined by an XML DTD (or other syntactic specification).
- Each metadatum, when considered in isolation from all other metadata, should have a value that is valid as defined by some specification. For instance, exposure times should be greater than zero.
- Metadata considered in pairs, triples, …, and n-tuples should be consistent as a group. For instance, exposure time and shutter speed should be pair-wise consistent.
- All metadata should be correct.

The Flashpix image file format suggested processing rules, applicable on an element by element basis, for metadata. In the Flashpix rules, some metadata should be copied, some should be deleted, and some should be copied only if the processing application understood the meaning of the metadata (and deleted otherwise).

The DIG35 approach for metadata persistence builds on the Flashpix effort, but with the variation that the processing application marks the metadata to indicate the processing that has taken place. The processing application might be creating, updating, deleting, or copying the metadata. The following rules cover each of those cases:

1. Metadata must be copied except where rule 2 or rule 3 applies.
2. Metadata must be deleted when the processing application "knows" the metadata should be deleted.
3. Metadata must be updated when the processing application "knows" the correct new value.
4. When metadata is copied the previous timestamp must be copied.
5. When metadata is created or updated, the associated timestamp must be set to "now".
6. When an application (with or without human intervention) has determined that metadata is correct, the timestamp of the metadata must be set to "now".

These metadata persistence rules permit a relatively simple test in subsequent processing applications. If a metadatum has a timestamp that is the same as the last image modification time, then there is no reason to doubt that the metadatum is incorrect. Metadata with timestamp(s) earlier than the last image modification time have not been identified as correct in the context of subsequent image and metadata modifications. Processing applications may suspect such metadata to be incorrect.

Metadata correctness is often more relative and less absolute, however. The metadata in the file is in general the best available estimate of the true metadata. The following rules cover the use of the metadata by processing applications:

1. The processing applications must treat the metadata as reliable when the metadatum timestamp matches the last image modification time.
2. The processing application may attempt to determine consistency and validity when metadata timestamps are earlier than the last image modification time.
3. The processing application should treat the metadata as reliable when the only indication that metadata may be invalid or inconsistent is that the metadata timestamp is earlier than the last image modification time.

The metadata timestamps and the image modification time metadatum are discussed in the Annexes.

NOTE: It is left for further study to determine if processing applications need more information about the provenance of a particular metadatum. Processing applications could add attributes to the metadata elements to indicate that the processing application did or didn't understand the metadata. Similarly, processing applications could, identify metadata that it had created; identify that the metadatum was determined to be correct programmatically; or that a human operator deemed the metadata to be correct.

5.7 Intellectual Property Rights (IPR)

Intellectual Property Rights (IPR) is a general concept addressing the ownership and usage conditions of image content. Because the content of a file can be either created from scratch (original works) or built upon existing material (derivative works), the IPR may be different for each scenario. There are two basic categories of IPR, which are as follows: (1) Moral Rights (attached to the creator and non-transferable); and (2) Copyrights (conditional rights to copying, using, and exploiting content).

The World Intellectual Property Organization (WIPO) has stated[25][26] that, once IPR information is included in a file, then the information becomes an integral part of the file and must be conveyed without modification along with the file by all intermediate actors between creator and end user. This concept of conveying the IPR information is called “persistence.” Therefore, IPR data is important metadata, which is inherently associated with the image data. Because IPR data cannot be erased or modified inside of the file (if it is to comply with legal requirements), it may become necessary to either add new data or to update the existing data. For instance, if an image is created by “collage” (i.e., compound image of multiple original images), there should be a special set of metadata that gives the provenance of each pieces so that the IPR of each image is respected. Another instance is when a photographer who is attached to one agency leaves the first agency and attaches to another and is allowed by the first agency to take all of his photographs with him. Similarly, if an agency disappears and a new agency is created, then the IPR metadata must persist and reflect this change.

The original IPR data cannot be erased, but additional data can be added. Another way to eliminate this type of situation is to use IPR data as a link to a secured database, which is managed by a Trusted Third Party (TTP), such as a law firm. IPR data is updated in the TTP log files, which can be easily accessed using the link, which is inside of the file. Such would be the case if the only IPR data that is present in the file were a watermark, inserted inside the image data. By detecting and extracting the value of the watermark, one gets the link to the TTP database, where he can read the latest, updated information. Access to IPR information, however, can be restricted in certain conditions, such as when the information is classified as “confidential.” Obviously, for the same reason, updating the IPR information, which could be done by adding information and a timestamp, is also restricted to conditional access.

All IPR metadata should be determined with a view to extensions, according to the future operation of automatic billing and crediting of both the end user and the creator. These systems generally called Electronic Copyright Management Systems (ECMS) and /or Intellectual Property Management and Protection (IPMP) are not completely defined as of today, although their main input and output data are those stated within this document.

Lastly, IPR are of significant concern, because there are serious ramifications to an end user’s actions regarding images. Publication, alteration, modification, etc. can have profound legal consequences. Also, there may be a series of rights that originate from a single, original image. Accordingly, the following is a non-exhaustive list of discussion items related to IPR:

- How much of the IRP metadata actually interests the personal applications, the professional applications?
- What is the IPR importance of the operator for processing, editing, collaging etc.?
- What is the IPR importance of the operator for scanning, color-correcting, as well as all types of enhancements?
- What should be done if an image file does not have consistent IPR information?
- What should be done if various treaties or international agreements change the requirements for IPR?
- What happens when a licensing authority or contact no longer exists but is still referenced in the metadata?

ANNEXES

Annex A: Basic Image Parameter Metadata

A.1 Overview

This Annex defines basic image parameter metadata that contains generic information about the image, such as the image size and number of components. While this metadata that are commonly referred to as the “Header Data” of an image file format, the scope of this Annex is to define metadata that is file format independent. Thus this metadata should be considered informational and not to be used to decode the image data stored in the associated file. The metadata fields would be considered secondary to the image data.

The metadata defined in this Annex is useful where the metadata is stored externally to the image file. For example, the number of pixels may be determined without interrogating the image file. In files such as the Flashpix file format where the fields defined in this section are clearly defined in the file specification, it would not be expected that the metadata fields would be stored. This is opposite to a file such as a BMP where this metadata is not defined and would be useful.

A.2 Structure

The following diagram illustrates the logical structure of the Basic Image Parameter Metadata.

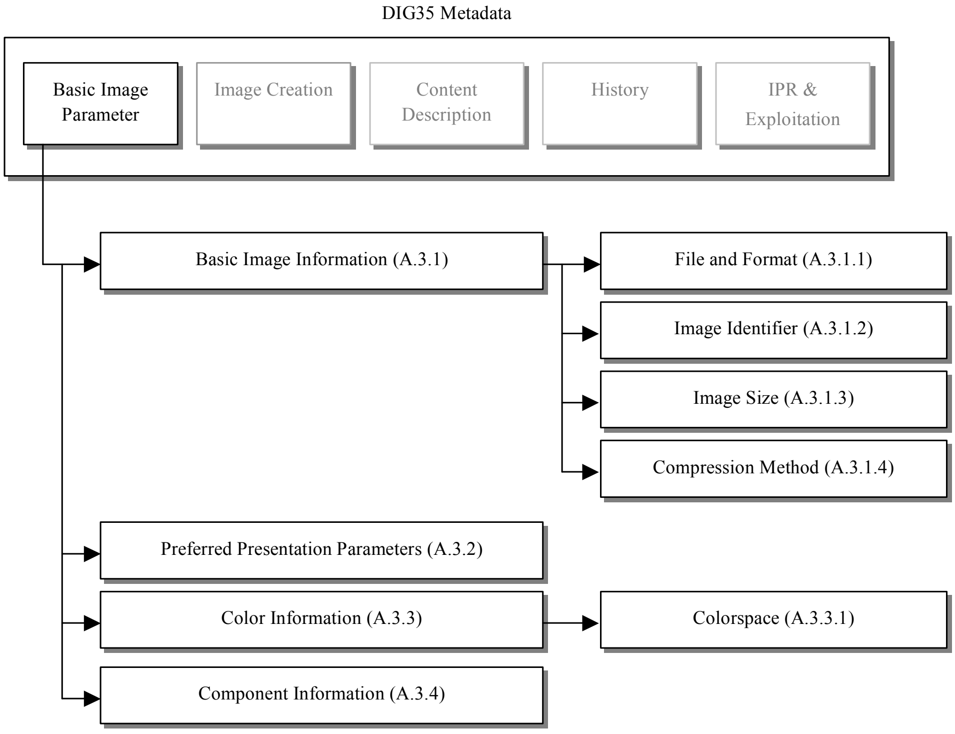

DIG35 Metadata
Basic Image Parameter
Image Creation
Content Description
History
IPR & Exploitation
Basic Image Information (A.3.1)
File and Format (A.3.1.1)
Image Identifier (A.3.1.2)
Image Size (A.3.1.3)
Compression Method (A.3.1.4)
Preferred Presentation Parameters (A.3.2)
Color Information (A.3.3)
Colorspace (A.3.3.1)
Component Information (A.3.4)

Figure A-1: Basic Image Parameter metadata structure

A.3 Definition

Basic Image Parameter Metadata may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="BASIC_IMAGE_PARAM">
  <xsd:complexType>
    <xsd:element ref="dig35:BASIC_IMAGE_INFO"                 minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="dig35:PREF_PRESENTATION_PARAM"          minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="dig35:COLOR_INFO"                       minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="dig35:COMPONENT_INFO"                   minOccurs="0"  maxOccurs="1"/>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

Basic Image Information: This field specifies generic information about the image. See section A.3.1 for details.

Preferred Presentation Parameters: This field specifies the preferred height and width, respectively for presentation purposes. See section A.3.2 for details.

Color Information: This field specifies the colorspace of the decompressed image data. See section A.3.3 for details.

Component information: This field contains information about the image data components. See section A.3.4 for details.

A.3.1 Basic Image Information

Basic Image Information specifies generic information about the image, such as the image size and number of components. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="BASIC_IMAGE_INFO">
  <xsd:complexType>
    <xsd:element ref="dig35:FILE_FORMAT"                      minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="dig35:IMAGE_ID"                         minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="dig35:IMAGE_SIZE"                       minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="dig35:COMPRESSION"                      minOccurs="0"  maxOccurs="1"/>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

File and Format: This field specifies the file name and format of the image. See section A.3.1.1 for details.

Image Identifier: This field specifies a unique identifier of the image. See section A.3.1.2 for details.

Image Size: This field specifies the width and height of the image. See section A.3.1.3 for details.

Compression Method: This field specifies the compression method used to store the image data. See section A.3.1.4 for details.

A.3.1.1 File and Format

File and Format specifies the format and name of the image file associated with the metadata. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="FILE_FORMAT">
  <xsd:complexType>
    <xsd:element name="FILE_NAME"    type="xsd:uriReference"  minOccurs="0"  maxOccurs="1"/>
    <xsd:element name="FORMAT_TYPE"  type="xsd:string"        minOccurs="0"  maxOccurs="1"/>
    <xsd:element name="MIME_TYPE"    type="xsd:string"        minOccurs="0"  maxOccurs="1"/>
    <xsd:element name="VERSION"      type="xsd:string"        minOccurs="0"  maxOccurs="1"/>
  </xsd:complexType>
</xsd:element>
```

File Name: This field specifies the name of image file.

File Format Type: This field specifies the file format of the image.

Table A-1: Suggested File Format Type values

Values	Meaning
BMP	Bitmap format
Exif	Exchangeable Image File Format
Flashpix	Flashpix file format
GIF	Graphics Interchange Format
JFIF	JPEG File Interchange Format
JP2	JPEG 2000 file format
PICT	PICT file format
PNG	Portable Network Graphic format
SPIFF	Still Picture Interchange File Format
TIFF	Tag Image File Format and its variant

MIME Type: This field specifies the Internet media type of the image file.

Version: This field specifies the version of the file format.

```
Example:

<FILE_FORMAT>
  <FILE_NAME>image.jpg</FILE_NAME>
  <FORMAT_TYPE>JFIF</FORMAT_TYPE>
  <VERSION>1.02</VERSION>
</FILE_FORMAT>
```

A.3.1.2 Image Identifier

Image Identifier specifies an image identifier that must uniquely identify the image(s) which bear them. The format may be globally unique (e.g. UUID), vendor or application dependent. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="IMAGE_ID">
  <xsd:complexType>
    <xsd:element name="UID"          type="xsd:string"        minOccurs="0"  maxOccurs="1"/>
    <xsd:element name="ID_TYPE"      type="xsd:uriReference"  minOccurs="0"  maxOccurs="1"/>
  </xsd:complexType>
</xsd:element>
```

Unique Identifier: This field specifies the unique identifier of an image. The Unique Identifier Type specifies the format of the field.

Unique Identifier Type: This field specifies the type of the Unique Identifier.

The Digital Imaging Group defined the following value for this field:

"http://www.digitalimaging.org/dig35/UUID"

This value specifies that the contents of this field in a UUID as defined by ISO/IEC 11578 [21]. Other vendors may define other values.

```
Example:

<IMAGE_ID>
  <UID>EC340B04-74C5-11D4-A729-879EA3548F0E</UID>
  <ID_TYPE> http://www.digitalimaging.org/dig35/UUID</ID_TYPE>
</IMAGE_ID>
```

A.3.1.3 Image Size

Image Size specifies the size of the image. For multiple-resolution image file formats, it shall specify the highest resolution. This field shall contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="IMAGE_SIZE"       type="dig35:tIntSize"/>
```

Width: This specifies the width of the image, in pixels.

Height: This specifies the height of the image, in pixels.

```
Example:

<IMAGE_SIZE>
  <WIDTH>1600</WIDTH>
  <HEIGHT>1200</HEIGHT>
</IMAGE_SIZE>
```

A.3.1.4 Compression Method

Compression Method specifies the compression method used to store the image data.

```
Schema Definition:

<xsd:element name="COMPRESSION"      type="xsd:string"/>
```

Table A-2: Suggested Compression Method values

Values	Meaning
JPEG	JPEG compression used
JPEG 2000	JPEG 2000 compression used
LZW	LZW compression used
LZ77	LZ77 compression and its variant used
RLE	Run Length Encoding used
Uncompressed	No compression used

```
Example:

<COMPRESSION>JPEG</COMPRESSION>
```

A.3.2 Preferred Presentation Parameters

Preferred Presentation Parameters specifies the preferred height and width, respectively. This size is the size that the image should be displayed on an output device when the output size is not specified by other means. For example, when printing the image directly, it should be shown this size. When printing the same image from a page layout application, then the page layout application may well specify the position and size of the image. (Note the preferred presentation parameters may be used as the initial size when the image is loaded into the page layout application.)

This field shall contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="PREF_PRESENTATION_PARAM"                   type="dig35:tDoubleSize"/>
```

Width: This field specifies the preferred width of the image in meters.

Height: This field specifies the preferred height of the image in meters.

```
Example:

<PREF_PRESENTATION_PARAM>
  <WIDTH>0.1524</WIDTH>
  <HEIGHT>0.1016</HEIGHT>
</PREF_PRESENTATION_PARAM>
```

A.3.3 Color Information

Color Information specifies the colorspace of the decompressed image data. Note that internal to the compression process, the image coder may convert the data to a different colorspace. However, this process is considered a black box at the file format level, and that internal colorspace is not exposed to the end user.

This field may contain the sub-fields listed below unless otherwise stated.

```
Schema Definition:

<xsd:element name="COLOR_INFO">
  <xsd:complexType>
    <xsd:element ref="dig35:COLORSPACE"                         maxOccurs="unbounded"/>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

Colorspace: This field must be included and specifies the colorspace. See section A.3.3.1 for details.

A.3.3.1 Colorspace

Colorspace specifies the colorspace of the decompressed image data as an ICC profile[4]. The profile may be specified either by a profile name or an URL. If it is named, then the profile is known by a well-defined name and thus is not included in the file; the name of the desired profile should be specified. If the value is `URL`, then the desired profile is located at the `URL`. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="COLORSPACE">
  <xsd:complexType>
    <xsd:element name="PROFILE_NAME" type="dig35:tLangString" minOccurs="0"   maxOccurs="1"/>
    <xsd:element name="PROFILE_REF"  type="xsd:uriReference"  minOccurs="0"   maxOccurs="1"/>
  </xsd:complexType>
</xsd:element>
```

Profile Name: This field specifies the well-known name of the ICC profile that specifies the colorspace of the decompressed image data.

Table A-3: Suggested Profile Name values

Values	Meaning
Grayscale	Grayscale colorspace used
sRGB	sRGB colorspace used
YCbCr	YCbCr colorspace used

Profile Reference: This field specifies the location of the ICC profile that specifies the colorspace of the decompressed image data.

```
Example:

<COLOR_INFO>
  <COLORSPACE>
    <PROFILE_NAME>sRGB</PROFILE_NAME>
  </COLORSPACE>
</COLOR_INFO>
```

A.3.4 Component Information

Component Information specifies information about the image data components. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="COMPONENT_INFO">
  <xsd:complexType>
    <xsd:element name="NUM_COMPONENT" type="xsd:positiveInteger"  minOccurs="0" maxOccurs="1"/>
    <xsd:element name="PREMULTIPLIED" type="xsd:boolean"          minOccurs="0" maxOccurs="1"/>
    <xsd:element name="COMPONENTS"    type="xsd:string"           minOccurs="0" maxOccurs="1"/>
    <xsd:element name="COMP_SIZE"     type="xsd:positiveInteger" minOccurs="0" maxOccurs="unbounded"/>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

Number of Components: This field specifies the number of components in the image. The value of this field is the same as the number of Component Size fields contained in the Component Information field.

Premultiplied: This field specifies whether the opacity component of the image has been premultiplied into the color components. If the value of this field is `true`, then the opacity component has been premultiplied into all of the other components. If the value of this field is `false`, then the opacity component has not been premultiplied into any components.

Component Description: This field contains a description of the components in the image.

Table A-4: Suggested Component Description values

Values	Meaning
RGB	Red, Green and Blue
RGBA	Red, Green, Blue and Alpha
CMY	Cyan, Magenta, Yellow
CMYK	Cyan, Magenta, Yellow, Black

Component Size: This field specifies the bit-depth of each component. Where there is a single component size value, it is the bit-depth of all components in the image. Where there are multiple values, the order of the fields is the same as the order of the components in the image data

```
Examples:

<!--
   - Example 1: 4 components, premultiplied, RGBA, all 8-bits
  -->
<COMPONENT_INFO>
  <NUM_COMPONENT>4</NUM_COMPONENT>
  <PREMULTIPLIED>true</PREMULTIPLIED>
  <COMPONENTS>RGBA</COMPONENTS>
  <COMP_SIZE>8</COMP_SIZE>
</COMPONENT_INFO>

<!--
   - Example 2: 3 components, YCbCr, 4:2:2
  -->
<COMPONENT_INFO>
  <NUM_COMPONENT>3</NUM_COMPONENT>
  <COMPONENTS>YCbCr</COMPONENTS>
  <COMP_SIZE>4</COMP_SIZE>
  <COMP_SIZE>2</COMP_SIZE>
  <COMP_SIZE>2</COMP_SIZE>
</COMPONENT_INFO>
```

A.4 Example

Following example shows a JPEG file’s Basic Image Parameters in XML.

```
Example:

<--
  - File name: image.jpg; File format: JPEG/JFIF; Image size: 1600 x 1200;
  -->
<METADATA TYPE="Single">
   <BASIC_IMAGE_PARAM>
      <BASIC_IMAGE_INFO>
         <FILE_FORMAT>
            <FILE_NAME>image.jpg</FILE_NAME>
            <FORMAT_TYPE>JFIF</FORMAT_TYPE>
            <VERSION>1.02</VERSION>
         </FILE_FORMAT>
         <IMAGE_SIZE>
           <WIDTH>1600</WIDTH>
           <HEIGHT>1200</HEIGHT>
         </IMAGE_SIZE>
         <COMPRESSION>JPEG</COMPRESSION>
      </BASIC_IMAGE_INFO>
      <COLOR_INFO>
         <COLORSPACE>
            <PROFILE_NAME>YCbCr</PROFILE_NAME>
         </COLORSPACE>
      </COLOR_INFO>
      <COMPONENT_INFO>
         <NUM_COMPONENT>3</NUM_COMPONENT>
         <COMPONENTS>YCbCr</COMPONENTS>
         <COMP_SIZE>4</COMP_SIZE>
         <COMP_SIZE>2</COMP_SIZE>
         <COMP_SIZE>2</COMP_SIZE>
      </COMPONENT_INFO>
   </BASIC_IMAGE_PARAM>
</METADATA>
```

Annex B: Image Creation Metadata

B.1 Overview

This Annex defines metadata that are related to the *creation* of a digital image. The scope of this Annex is applicable to metadata fields that are relevant to the creation of the digital image data, i.e. camera and scanner device information and its capture condition as well as the software or firmware to create such image. It defines the "how" metadata that specifies the pedigree of the image. In general, creation metadata is considered to be "read-only" being persistent and associated with the image through the workflow.

B.2 Structure

The following diagram illustrates the logical structure of the Image Creation metadata.

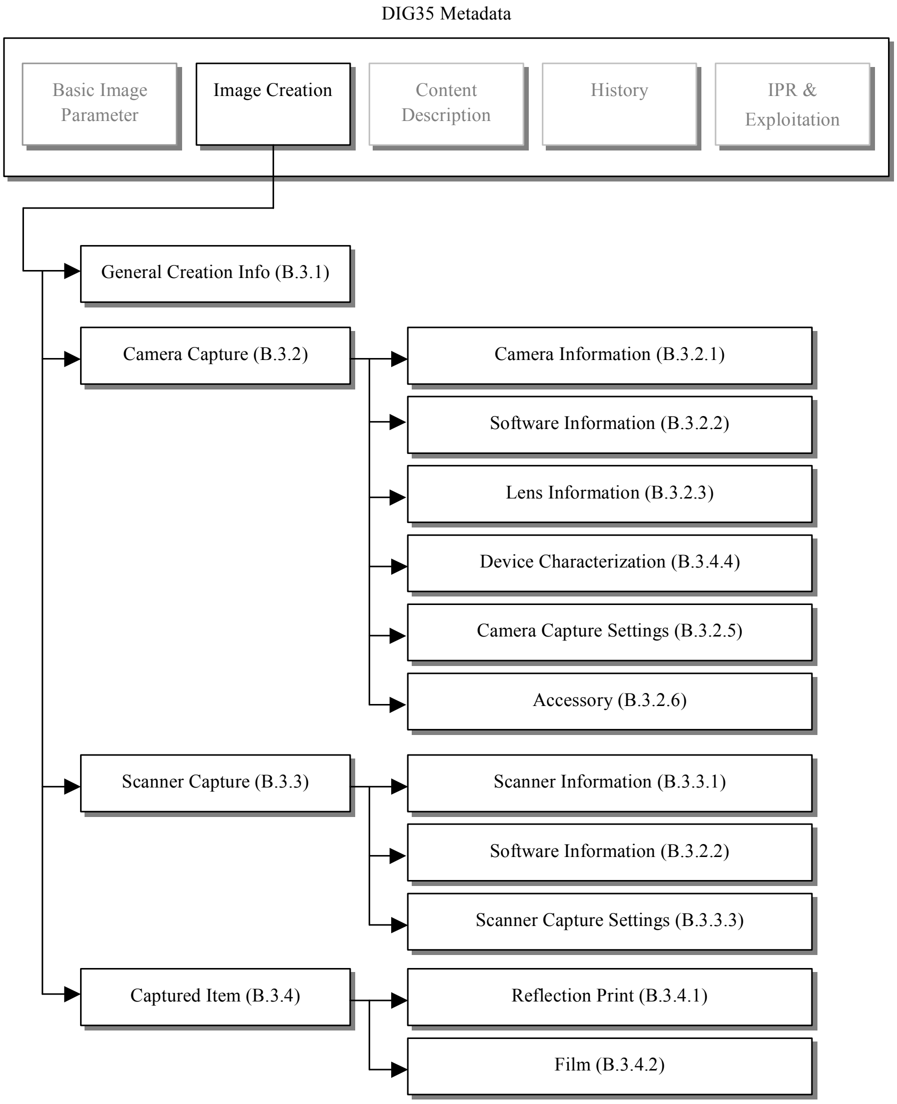

DIG35 Metadata
Basic Image Parameter
Image Creation
Content Description
History
IPR & Exploitation
General Creation Info (B.3.1)
Camera Capture (B.3.2)
Camera Information (B.3.2.1)
Software Information (B.3.2.2)
Lens Information (B.3.2.3)
Device Characterization (B.3.4.4)
Camera Capture Settings (B.3.2.5)
Accessory (B.3.2.6)
Scanner Capture (B.3.3)
Scanner Information (B.3.3.1)
Software Information (B.3.2.2)
Scanner Capture Settings (B.3.3.3)
Captured Item (B.3.4)
Reflection Print (B.3.4.1)
Film (B.3.4.2)

Figure B-1: Image Creation metadata structure

B.3 Definition

Image Creation Metadata may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="IMAGE_CREATION">
  <xsd:complexType>
    <xsd:element ref="dig35:GENERAL_CREATION_INFO"          minOccurs="0"   maxOccurs="1"/>
    <xsd:element ref="dig35:CAMERA_CAPTURE"                 minOccurs="0"   maxOccurs="1"/>
    <xsd:element ref="dig35:SCANNER_CAPTURE"                minOccurs="0"   maxOccurs="1"/>
    <xsd:element ref="dig35:CAPTURED_ITEM"                  minOccurs="0"   maxOccurs="1"/>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

General Creation Information: This field specifies generic information on how the image was created. See section B.3.1 for details.

Camera Capture: This field specifies a camera capture metadata of a scene. See section B.3.2 for details.

Scanner Capture: This field specifies scanner capture metadata that may be used for various scanners such as flatbed and film scanners. See section B.3.3 for details.

Captured Item: This field contains description of the item that was digitally captured. See section B.3.4 for details.

B.3.1 General Creation Information

General Creation Information specifies general information on how the image was created. Applications may choose to skip further parsing based on the values stored here. For example if the application is only interested in Camera Capture metadata, it can skip additional parsing based on the Image Source value. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="GENERAL_CREATION_INFO">
  <xsd:complexType>
    <xsd:element name="CREATION_TIME" type="xsd:timeInstant"     minOccurs="0"     maxOccurs="1"/>
    <xsd:element name="IMAGE_SOURCE"  type="dig35:tLangString"   minOccurs="0"     maxOccurs="1"/>
    <xsd:element name="SCENE_TYPE"    type="dig35:tLangString"   minOccurs="0"     maxOccurs="1"/>
    <xsd:element name="IMAGE_CREATOR" type="dig35:tPerson"       minOccurs="0"     maxOccurs="1"/>
    <xsd:element name="OPERATOR_ORG"  type="dig35:tOrganization" minOccurs="0"     maxOccurs="1"/>
    <xsd:element name="OPERATOR_ID"   type="dig35:tLangString"   minOccurs="0"     maxOccurs="1"/>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

Creation Time: This field specifies the date and time the image was created. This field should be stored when the creation process started. (E.g. it may be an 8-minute exposure.) This field should never be changed after it is written in the image creation device.

Image Source: This field specifies the device source of the digital file, such as a film scanner, reflection print scanner, or digital camera. Table B-1 lists suggested image source values.

Table B-1: Suggested Image Source values

Value	Meaning
`Digital Camera`	Image create by a digital camera
`Film Scanner`	Image create by a film scanner
`Reflection Print Scanner`	Image create by a reflection print scanner (commonly referred to as a “flat bed”)
`Still From Video`	Image create by from video
`Computer Graphics`	Image digitally created on computers

Scene Type: This field specifies the type of scene that was captured. It differentiates “original scenes” (direct capture of real-world scenes) from “second generation scenes” (images captured from pre-existing hardcopy images). It provides further differentiation for scenes that are digitally composed.

Table B-2: Suggested Scene Type values

Value	Meaning
`Original Scene`	Direct capture of real-world scenes
`Second Generation Scene`	Images captured from pre-existing hardcopy images such a photograph
`Digital Scene Generation`	Graphic arts or images digitally composed

Image Creator: This field specifies the name of the image creator. The image creator could be, for example, the photographer who captured the original picture on film, the illustrator, or graphic artist who conducted the image-creation process, etc. See section F.2.13 for the format of this field.

Operator Organization: This field specifies the name of the service bureau, photofinisher, or organization where the image capture process (photographed, scanned or created by software) is conducted. See section F.2.14 for the format of this field.

Operator ID: This field specifies a name or ID for the organization conducting the capture process.

```
Example:

<GENERAL_CREATION_INFO>
  <CREATION_TIME>2000-06-08T21:07:00</CREATION_TIME>
  <IMAGE_SOURCE>Film Scanner</IMAGE_SOURCE>
  <SCENE_TYPE>Second Generation Scene</SCENE_TYPE>
  <IMAGE_CREATOR>
    <PERSON_NAME>
      <NAME_COMP TYPE="Family">Brown</NAME_COMP>
    </PERSON_NAME>
  </IMAGE_CREATOR>
  <OPERATOR_ORG>
    <ORG_NAME>Acme Inc</ORG_NAME>
  </OPERATOR_ORG>
  <OPERATOR_ID>X23r7</OPERATOR_ID>
</GENERAL_CREATION_INFO>
```

B.3.2 Camera Capture

Camera Capture specifies a camera capture of a scene. It contains information regarding the camera, the lens, device characterization and camera capture settings. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="CAMERA_CAPTURE">
  <xsd:complexType>
    <xsd:element name="CAMERA_INFO"   type="dig35:tProductDetails" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="SOFTWARE_INFO" type="dig35:tProductDetails" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="LENS_INFO"     type="dig35:tProductDetails" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="dig35:DEVICE_CHARACTER"                      minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="dig35:CAMERA_SETTINGS"                       minOccurs="0" maxOccurs="1"/>
    <xsd:element name="ACCESSORY"     type="dig35:tProductDetails" minOccurs="0" maxOccurs="unbounded"/>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

B.3.2.1 Camera Information

Camera Information specifies information about the camera that captured the image. See section F.2.18 for the format of this field.

```
Example:

<CAMERA_INFO>
  <MANUFACTURER>
    <ORG_NAME>Acme</ORG_NAME>
  </MANUFACTURER>
  <MODEL>Model 1000</MODEL>
  <SERIAL>2941554</SERIAL>
</CAMERA_INFO>
```

B.3.2.2 Software Information

Software Information specifies information about the software or firmware used to capture the image. See section F.2.18 for the format of this field.

B.3.2.3 Lens Information

Lens Information specifies information about the lens that captured the image. See section F.2.18 for the format of this field.

```
Example:

<LENS_INFO>
  <MANUFACTURER>
    <ORG_NAME>Acme</ORG_NAME>
  </MANUFACTURER>
  <MODEL>35mm-105mm Zoom</MODEL>
  <SERIAL>4576354</SERIAL>
</LENS_INFO>
```

B.3.2.4 Device Characterization

Device Characterization specifies the technical characterization of the digital capture device. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="DEVICE_CHARACTER">
  <xsd:complexType>
    <xsd:element ref="SENSOR_TECHNOLOGY"                         minOccurs="0"    maxOccurs="1"/>
    <xsd:element name="FOCAL_PLANE_RES"    type="dig35:tIntSize" minOccurs="0"    maxOccurs="1"/>
    <xsd:element name="SPECTRAL_SENSITIVITY"                     minOccurs="0"    maxOccurs="1">
      <xsd:complexType>
        <xsd:any processContents="skip"/>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="ISO_SATURATION"     type="dig35:tNonNegativeDouble"
                                                                 minOccurs="0"    maxOccurs="1"/>
    <xsd:element name="ISO_NOISE"          type="dig35:tNonNegativeDouble"
                                                                 minOccurs="0"    maxOccurs="1"/>
    <xsd:element ref="SPATIAL_FREQ_RESPONSE"                     minOccurs="0"    maxOccurs="1"/>
    <xsd:element ref="CFA_PATTERN"                               minOccurs="0"    maxOccurs="1"/>
    <xsd:element ref="OECF"                                      minOccurs="0"    maxOccurs="1"/>
    <xsd:element name="MIN_F_NUMBER"       type="dig35:tNonNegativeDouble"
                                                                 minOccurs="0"    maxOccurs="1"/>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="SENSOR_TECHNOLOGY">
  <xsd:simpleType base="xsd:string">
    <xsd:enumeration value="One-Chip Color Area"/>
    <xsd:enumeration value="Two-Chip Color Area"/>
    <xsd:enumeration value="Three-Chip Color Area"/>
    <xsd:enumeration value="Color Sequential Area"/>
    <xsd:enumeration value="Trilinear"/>
    <xsd:enumeration value="Color Sequential Linear Sensor"/>
  </xsd:simpleType>
</xsd:element>

<xsd:element name="SPATIAL_FREQ_RESPONSE">
  <xsd:complexType>
    <xsd:element ref="SPATIAL_FREQ_VAL"                                       maxOccurs="unbounded"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="SPATIAL_FREQ_VAL">
  <xsd:complexType>
    <xsd:element name="SPATIAL_FREQ" type="dig35:tNonNegativeDouble"/>
    <xsd:element name="HORIZ_SFR"    type="dig35:tNonNegativeDouble"/>
    <xsd:element name="VERT_SFR"     type="dig35:tNonNegativeDouble"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="CFA_PATTERN">
  <xsd:complexType>
    <xsd:element ref="COLOR_ROW"                                              maxOccurs="unbounded"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="COLOR_ROW">
```

```
  <xsd:complexType>
    <xsd:element name="COLOR"                                                 maxOccurs="unbounded">
      <xsd:simpleType base="xsd:string">
        <xsd:enumeration value="Red"/>
        <xsd:enumeration value="Green"/>
        <xsd:enumeration value="Blue"/>
        <xsd:enumeration value="Cyan"/>
        <xsd:enumeration value="Magenta"/>
        <xsd:enumeration value="Yellow"/>
        <xsd:enumeration value="White"/>
      </xsd:simpleType>
    </xsd:element>
  </xsd:complexType>
</xsd:element>

<xsd:element name="OECF">
  <xsd:complexType>
    <xsd:element ref="LOG_VAL"                                                maxOccurs="unbounded"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="LOG_VAL">
  <xsd:complexType>
    <xsd:element name="LOG_EXPOSURE" type="xsd:double"/>
    <xsd:element name="OUTPUT_LEVEL" type="dig35:tNonNegativeDouble"          maxOccurs="unbounded"/>
  </xsd:complexType>
</xsd:element>
```

Sensor Technology: This field specifies either the type of image sensor or the sensing method used in the camera or image-capture device.

Focal Plane Resolution: This field specifies the number of pixels in the X (width) and Y (height) directions for the main image. The width and height stored are the width and height of the image generated rather than the width and height of the image sensor.

Spectral Sensitivity: This field can be used to describe the spectral sensitivity of each channel of the camera used to capture the image. It is useful for certain scientific applications. The content of this field is compatible with ASMT E1708-95 [2] and expected to be defined by another standard.

ISO Saturation Speed Rating: This field specifies the ISO saturation speed rating classification as defined in ISO 12232 [14].

ISO Noise Speed Rating: This field specifies the ISO noise-based speed rating classification as defined in ISO 12232.

Spatial Frequency Response: This specifies the spatial frequency response (SFR) of the image capture device. The device-measured SFR data, described in ISO 12233 [15], can be stored as a table of spatial frequencies, horizontal SFR values, vertical SFR values, and diagonal SFR values. The following is a simple example of measured SFR data table.

Table B-3: Example Spatial Frequency Response value

Spatial frequency (lw/ph [i])	Horizontal SFR	Vertical SFR
0.1	1.00	1.00
0.2	0.90	0.95
0.3	0.80	0.85

CFA Pattern: Encodes the actual color filter array (CFA) geometric pattern of the image sensor used to capture a single-sensor color image. It is not relevant for all sensing methods. The data contains the minimum number of rows and columns of filter color values that uniquely specify the color filter array. The following table shows a simple example of a CFA Pattern, surrounded in bold lines, in a color filter array. Table B-4 shows a sample CFA Pattern.

[i] line widths per picture height

Table B-4: Sample CFA Pattern

Green	Red	Green	Red	…
Blue	Green	Blue	Green	…
Green	Red	Green	Red	…
Blue	Green	Blue	Green	…
…	…	…	…	…

OECF: This field specifies the opto-electronic conversion function (OECF). The OECF is the relationship between the optical input and the image file code value outputs of an electronic camera. The property allows OECF values defined in ISO 14524 [17] to be stored as a table of values. The following table shows a simple example of measured OECF data.

Table B-5: An example of measured OECF data

Log exposure	Red output level	Green output level	Blue output level
-3.0	10.2	12.5	8.9
-2.0	48.1	47.5	48.3
-1.0	150.2	152.0	149.8

Minimum F-Number: This field specifies the minimum lens f-number of the camera or image capture device.

```
Example:

<DEVICE_CHARACTER>
  <SENSOR_TECHNOLOGY>Trilinear</SENSOR_TECHNOLOGY>

  <FOCAL_PLANE_RES>
    <WIDTH>1600</WIDTH>
    <HEIGHT>1200</HEIGHT>
  </FOCAL_PLANE_RES>

  <ISO_SATURATION>40</ISO_SATURATION>
  <ISO_NOISE>40</ISO_NOISE>

  <SPATIAL_FREQ_RESPONSE>
    <SPATIAL_FREQ_VAL>
      <SPATIAL_FREQ>0.1</SPATIAL_FREQ>
      <HORIZ_SFR>1.0</HORIZ_SFR>
      <VERT_SFR>1.0</VERT_SFR>
    </SPATIAL_FREQ_VAL>
    <SPATIAL_FREQ_VAL>
      <SPATIAL_FREQ>0.2</SPATIAL_FREQ>
      <HORIZ_SFR>0.9</HORIZ_SFR>
      <VERT_SFR>0.95</VERT_SFR>
    </SPATIAL_FREQ_VAL>
    <SPATIAL_FREQ_VAL>
      <SPATIAL_FREQ>0.3</SPATIAL_FREQ>
      <HORIZ_SFR>0.8</HORIZ_SFR>
      <VERT_SFR>0.85</VERT_SFR>
    </SPATIAL_FREQ_VAL>
  </SPATIAL_FREQ_RESPONSE>

  <CFA_PATTERN>
    <COLOR_ROW>
      <COLOR>Green</COLOR>
      <COLOR>Red</COLOR>
    </COLOR_ROW>
    <COLOR_ROW>
      <COLOR>Blue</COLOR>
      <COLOR>Green</COLOR>
    </COLOR_ROW>
  </CFA_PATTERN>

  <OECF>
    <LOG_VAL>
      <LOG_EXPOSURE>-3.0</LOG_EXPOSURE>
      <OUTPUT_LEVEL>10.2</OUTPUT_LEVEL>
      <OUTPUT_LEVEL>12.5</OUTPUT_LEVEL>
      <OUTPUT_LEVEL>8.9</OUTPUT_LEVEL>
    </LOG_VAL>
    <LOG_VAL>
      <LOG_EXPOSURE>-2.0</LOG_EXPOSURE>
      <OUTPUT_LEVEL>48.1</OUTPUT_LEVEL>
      <OUTPUT_LEVEL>47.5</OUTPUT_LEVEL>
      <OUTPUT_LEVEL>48.3</OUTPUT_LEVEL>
    </LOG_VAL>
    <LOG_VAL>
      <LOG_EXPOSURE>-1.0</LOG_EXPOSURE>
      <OUTPUT_LEVEL>150.2</OUTPUT_LEVEL>
      <OUTPUT_LEVEL>152.0</OUTPUT_LEVEL>
      <OUTPUT_LEVEL>149.8</OUTPUT_LEVEL>
    </LOG_VAL>
  </OECF>

  <MIN_F_NUMBER>1.4</MIN_F_NUMBER>
</DEVICE_CHARACTER>
```

B.3.2.5 Camera Capture Settings

Camera Capture Settings specify the camera settings used when the image was captured. New generations of digital and film cameras make it possible to capture more information about the conditions under which a picture was taken. This may include information about the lens aperture and exposure time, whether a flash was used, which lens was used, etc. This technical information is useful to professional and serious amateur photographers. In addition, some of these properties are useful to image database applications for populating values useful to advanced imaging applications and algorithms as well as image analysis and retrieval. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="CAMERA_SETTINGS">
  <xsd:complexType>
    <xsd:choice                                                    minOccurs="0" maxOccurs="1">
      <xsd:element name="EXP_TIME"        type="dig35:tNonNegativeDouble"/>
      <xsd:element name="R_EXP_TIME"      type="dig35:tRational"/>
    </xsd:choice>

    <xsd:element name="F_NUMBER" type="dig35:tNonNegativeDouble"   minOccurs="0"  maxOccurs="1"/>
    <xsd:element name="EXP_PROGRAM"       type="dig35:tLangString" minOccurs="0"  maxOccurs="1"/>
    <xsd:element name="BRIGHTNESS"        type="xsd:double"        minOccurs="0"  maxOccurs="1"/>
    <xsd:element name="EXPOSURE_BIAS"     type="xsd:double"        minOccurs="0"  maxOccurs="1"/>
    <xsd:element name ="SUBJECT_DISTANCE" type="dig35:tNonNegativeDouble"
                                                                   minOccurs="0"  maxOccurs="1"/>
    <xsd:element name ="METERING_MODE"    type="dig35:tLangString" minOccurs="0"  maxOccurs="1"/>
    <xsd:element name ="SCENE_ILLUMINANT"
                                          type="dig35:tLangString" minOccurs="0"  maxOccurs="1"/>
    <xsd:element name="COLOR_TEMP"        type="dig35:tNonNegativeDouble"
                                                                   minOccurs="0"  maxOccurs="1"/>
    <xsd:element name="FOCAL_LENGTH"      type="dig35:tNonNegativeDouble"
                                                                   minOccurs="0"  maxOccurs="1"/>
    <xsd:element name="FLASH"             type="xsd:boolean"       minOccurs="0"  maxOccurs="1"/>
    <xsd:element name="FLASH_ENERGY"      type="dig35:tNonNegativeDouble"
                                                                   minOccurs="0"  maxOccurs="1"/>
    <xsd:element name="FLASH_RETURN"      type="xsd:boolean"       minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="BACK_LIGHT"                                  minOccurs="0"  maxOccurs="1"/>
    <xsd:element name="SUBJECT_POSITION"  type="dig35:tPosition"   minOccurs="0"  maxOccurs="1"/>
    <xsd:element name="EXPOSURE_INDEX"    type="xsd:double"        minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="AUTO_FOCUS"                                  minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="SPECIAL_EFFECT"                              minOccurs="0"  maxOccurs="unbounded"/>
    <xsd:element name="CAMERA_LOCATION"   type="dig35:tLocation"   minOccurs="0"  maxOccurs="1"/>
    <xsd:element name="ORIENTATION"       type="dig35:tDirection"  minOccurs="0"  maxOccurs="1"/>
    <xsd:element name="PAR"               type="dig35:tRational"   minOccurs="0"  maxOccurs="1"/>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="BACK_LIGHT">
  <xsd:simpleType base="xsd:string">
    <xsd:enumeration value="Front Light"/>
    <xsd:enumeration value="Back Light 1"/>
    <xsd:enumeration value="Back Light 2"/>
  </xsd:simpleType>
</xsd:element>

<xsd:element name="AUTO_FOCUS">
  <xsd:simpleType base="xsd:string">
    <xsd:enumeration value="Auto Focus Used"/>
    <xsd:enumeration value="Auto Focus Interrupted"/>
    <xsd:enumeration value="Near Focused"/>
    <xsd:enumeration value="Soft Focused"/>
    <xsd:enumeration value="Manual"/>
  </xsd:simpleType>
</xsd:element>

<xsd:element name="SPECIAL_EFFECT">
  <xsd:simpleType base="xsd:string">
    <xsd:enumeration value="Colored"/>
    <xsd:enumeration value="Diffusion"/>
    <xsd:enumeration value="Multi-Image"/>
    <xsd:enumeration value="Polarizing"/>
    <xsd:enumeration value="Split-Field"/>
    <xsd:enumeration value="Star"/>
  </xsd:simpleType>
</xsd:element>
```

Exposure Time: This field specifies the exposure time used when the image was captured. The value of this field is stored in seconds.

F-Number: This field specifies the lens f-number (ratio of lens aperture to focal length) used when the image was captured.

Exposure Program: This field specifies the class of exposure program that the camera used at the time the image was captured.

Table B-6: Suggested Exposure Program values

Value	Meaning
`Manual`	The exposure setting set manually by the photographer.
`Program Normal`	A general purpose auto-exposure program
`Aperture Priority`	The user selected the aperture and the camera selected the shutter speed for proper exposure
`Shutter Priority`	The user selected the shutter speed and the camera selected the aperture for proper exposure
`Program Creative`	The exposure setting is biased toward greater depth of field
`Program Action`	The exposure setting is biased toward faster shutter speed
`Portrait Mode`	The exposure setting is intended for close-up photos with the background out of focus
`Landscape Mode`	The exposure setting is intended for landscapes with the background in good focus

Brightness Value: This field specifies the Brightness Value (Bv) measured when the image was captured, using APEX units. The expected maximum value is approximately 13.00 corresponding to a picture taken of a snow scene on a sunny day, and the expected minimum value is approximately –3.00 corresponding to a night scene. If the value supplied by the capture device represents a range of values rather than a single value, the minimum and maximum value may be specified.

Exposure Bias Value: This field specifies the actual exposure bias (the amount of over or under-exposure relative to a normal exposure, as determined by the camera's exposure system) used when capturing the image, using APEX units. The value is the number of exposure values (stops). For example, -1.00 indicates 1 eV (1 stop) underexposure, or half the normal exposure.

Subject Distance: This field specifies the distance between the front nodal plane of the lens and the subject on which the camera was focusing. Note that the camera may have focused on a subject within the scene that may not have been the primary subject. The subject distance may be specified by a single number if the exact value is known. Alternatively, a range of values indicating the minimum and maximum distance of the subject may be set. The value of this field is in meters.

Metering Mode: This field specifies the metering mode (the camera's method of spatially weighting the scene luminance values to determine the sensor exposure) used when capturing the image.

Table B-7: Suggested Metering Mode values

Value
Average
Center Weighted Average
Spot
MultiSpot
Pattern
Partial

Scene Illuminant: This field specifies the light source (scene illuminant) that was present when the image was captured.

Table B-8: Suggested Scene Illuminant values

Value
Daylight
Fluorescent Light
Tungsten Lamp
Flash
Standard Illuminant A
Standard Illuminant B
Standard Illuminant C
D55 Illuminant
D65 Illuminant
D75 Illuminant

Color Temperature: This field specifies the actual color temperature value of the scene illuminant stored in units of Kelvin.

Focal Length: This field specifies the lens focal length used to capture the image. Both a fixed lens focal length and a zoom lens focal length may be specified by a single number. To specify the zoom lens focal length you must know the zoom position. The value of this field is stored in meters.

Flash: This field specifies whether flash was used at image capture.

Flash Energy: This field specifies the amount of flash energy that was used. The measurement units are Beam Candle Power Seconds (BCPS).

Flash Return: This field specifies whether the camera judged that the flash was not effective at the time of exposure.

Back Light: This field specifies the camera's evaluation of the lighting conditions at the time of exposure.

Table B-9: Back Light values

Value	Meaning
`Front Light`	The subject is illuminated from the front side.
`Back Light 1`	The brightness value difference between the subject center and the surrounding area is greater than one full step (APEX). The frame is exposed for the subject center.
`Back Light 2`	The brightness value difference between the subject center and the surrounding area is greater than one full step (APEX). The frame is exposed for the surrounding area.

Subject Position: This field specifies the approximate position of the subject in the scene. See section F.2.17 for the format of this field.

Exposure Index: This field specifies the exposure index setting the camera selected.

Auto Focus: This field specifies the status of the focus of the capture device at the time of capture.

Table B-10: Auto Focus values

Value	Meaning
`Auto Focus Used`	The camera successfully focused on the subject.
`Auto Focus Interrupted`	The image was captured before the camera had successfully focused on the subject.
`Near Focused`	The camera deliberately focused at a distance closer than the subject to allow for the super-imposition of a focused foreground subject.
`Soft Focused`	The camera deliberately did not focus exactly at the subject distance to create a softer image (commonly used for portraits).
`Manual`	The camera was focused manually.

Special Effects: This field specifies the types of special effects filters used. It contains a list of filter fields, where the order of the fields in the array indicates the stacking order of the filters. The first value in the array is the filter closest to the original scene.

Camera Location: This field specifies the location of the camera when the picture was taken. See section F.2.15 for the format of this field.

Orientation: This field specifies the orientation of the camera when the picture was taken. See section F.2.16 for the format of this field.

Print Aspect Ratio (PAR): This field specifies the print aspect ratio specified by the user when the picture was taken.

```
Example:

<CAMERA_SETTINGS>

  <R_EXP_TIME>1/125</R_EXP_TIME> <!-- Exposure time as a Rational -->
  <F_NUMBER>1.4</F_NUMBER>
  <EXP_PROGRAM>Manual</EXP_PROGRAM>
  <BRIGHTNESS>-3.0</BRIGHTNESS>
  <EXPOSURE_BIAS>-1.0</EXPOSURE_BIAS>
  <SUBJECT_DISTANCE>10.5</SUBJECT_DISTANCE>
  <METERING_MODE>Average</METERING_MODE>
  <SCENE_ILLUMINANT>Flash</SCENE_ILLUMINANT>
  <FOCAL_LENGTH>0.005</FOCAL_LENGTH>
  <FLASH>true</FLASH>
  <FLASH_RETURN>false</FLASH_RETURN>
  <BACK_LIGHT>Front Light</BACK_LIGHT>

  <SUBJECT_POSITION>
```

```
    <RECT>
      <X>0.5</X>
      <Y>0.5</Y>
      <WIDTH>0.1</WIDTH>
      <HEIGHT>0.1</HEIGHT>
    </RECT>
  </SUBJECT_POSITION>

  <EXPOSURE_INDEX>50</EXPOSURE_INDEX>
  <AUTO_FOCUS>Auto Focus Used</AUTO_FOCUS>

  <SPECIAL_EFFECT>Colored</SPECIAL_EFFECT>
  <SPECIAL_EFFECT>Multi-Image</SPECIAL_EFFECT>

  <CAMERA_LOCATION>
    <COMMENT>Attached to head.</COMMENT>
  </CAMERA_LOCATION>

  <ORIENTATION>
    <YAW>0</YAW>
    <PITCH>0</PITCH>
    <ROLL>0</ROLL>
  </ORIENTATION>

  <PAR>16/9</PAR>

</CAMERA_SETTINGS>
```

B.3.2.6 Accessory

Accessory specifies accessories used with the capture device. Professional and amateur photographers may want to keep track of a variety of miscellaneous technical information, such as the use of extension tubes, bellows, close-up lenses, and other specialized accessories. See section F.2.18 for the format of this field.

```
Examples:

<ACCESSORY>
  <MODEL>Tripod</MODEL>
</ACCESSORY>
<ACCESSORY>
  <MODEL>Flash</MODEL>
</ACCESSORY>
```

B.3.3 Scanner Capture

Scanner Capture specifies scanner capture metadata that may be used for various scanners such as flatbed and film scanners. It optionally contains scanner information, device characterization and scanner capture settings. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="SCANNER_CAPTURE">
 <xsd:complexType>
    <xsd:element name="SCANNER_INFO" type="dig35:tProductDetails" minOccurs="0"   maxOccurs="1"/>
    <xsd:element name="SOFTWARE_INFO"type="dig35:tProductDetails" minOccurs="0"   maxOccurs="1"/>
    <xsd:element ref="dig35:SCANNER_SETTINGS"                     minOccurs="0"   maxOccurs="1"/>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

B.3.3.1 Scanner Information

Scanner Information specifies information about a particular scanner that was used to digitize an image item. It is recommended that applications are able to create a unique value of the scanner by combining all fields. See section F.2.18 for the format of this field.

B.3.3.2 Software Information

Software Information specifies information about the software or firmware used to capture the image. See section F.2.18 for the format of this field.

```
Example:

<SCANNER_CAPTURE>
  <SCANNER_INFO>
    <MANUFACTURER>
      <ORG_NAME>Acme</ORG_NAME>
    </MANUFACTURER>
  </SCANNER_INFO>
  <SOFTWARE_INFO>
    <MANUFACTURER>
      <ORG_NAME>Acme</ORG_NAME>
    </MANUFACTURER>
  </SOFTWARE_INFO>
</SCANNER_CAPTURE>
```

B.3.3.3 Scanner Capture Settings

Scanner Capture Settings specifies the scanner settings used when the image was scanned. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="SCANNER_SETTINGS">
  <xsd:complexType>
    <xsd:element name="PIXEL_SIZE"         type="dig35:tNonNegativeDouble"  minOccurs="0" maxOccurs="1"/>
    <xsd:element name="PHYSICAL_SCAN_RES"  type="dig35:tDoubleSize"         minOccurs="0" maxOccurs="1"/>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
  </xsd:complexType>
</xsd:element>
```

Pixel Size: This field specifies the pixel size, in meters, of the scanner.

Physical Scan Resolution: These field specify the physical scanning resolution of the device (not the interpolated resolution of the final output data) in the X (width) and Y (height) directions. The values of these fields are in meters.

```
Example:

<SCANNER_SETTINGS>
  <PIXEL_SIZE>0.000106</PIXEL_SIZE>
  <PHYSICAL_SCAN_RES>
    <WIDTH>0.21</WIDTH>
    <HEIGHT>0.297</HEIGHT>
  </PHYSICAL_SCAN_RES>
</SCANNER_SETTINGS>
```

B.3.4 Captured Item

Captured Item specifies capture item metadata. This field must contain either a Reflection Print or a Film. This field may contain the other sub-fields listed below.

```
Schema Definition:

<xsd:element name="CAPTURED_ITEM">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element ref="dig35:REFLECTION_PRINT"/>
      <xsd:element ref="dig35:FILM"/>
    </xsd:choice>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

Reflection Print: This field specifies information about the reflection print. See section B.3.4.1 for details.

Film: This field specifies information about the film. See section B.3.4.2 for details.

B.3.4.1 Reflection Print

Reflection Print specifies information about a reflection print that was digitally captured. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="REFLECTION_PRINT">
  <xsd:complexType>
    <xsd:element name="DOCUMENT_SIZE"  type="dig35:tDoubleSize"   minOccurs="0"    maxOccurs="1"/>
    <xsd:element ref="MEDIUM"                                     minOccurs="0"    maxOccurs="1"/>
    <xsd:element ref="RP_TYPE"                                    minOccurs="0"    maxOccurs="1"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="MEDIUM">
  <xsd:simpleType base="xsd:string">
    <xsd:enumeration value="Continuous Tone Image"/>
    <xsd:enumeration value="Halftone Image"/>
    <xsd:enumeration value="Line Art"/>
  </xsd:simpleType>
</xsd:element>

<xsd:element name="RP_TYPE">
  <xsd:simpleType base="xsd:string">
    <xsd:enumeration value="B/W Print"/>
    <xsd:enumeration value="Color Print"/>
    <xsd:enumeration value="B/W Document"/>
    <xsd:enumeration value="Color Document"/>
  </xsd:simpleType>
</xsd:element>
```

Document Size: This field specifies the lengths of the X (width) and Y (width) dimension of the original photograph or document, respectively. The values of these fields are given in meters.

Medium: This field specifies the medium of the original photograph, document, or artifact.

Reflection Print Type: This field specifies the type of the original document or photographic print.

```
Example:

<REFLECTION_PRINT>
  <DOCUMENT_SIZE>
    <WIDTH>0.21</WIDTH>
    <HEIGHT>0.297</HEIGHT>
  </DOCUMENT_SIZE>
  <MEDIUM>Line Art</MEDIUM>
  <RP_TYPE>B/W Document</RP_TYPE>
</REFLECTION_PRINT>
```

B.3.4.2 Film

Film specifies information on the film that was digitized. This field may contain the other sub-fields listed below.

```
Schema Definition:

<xsd:element name="FILM">
  <xsd:complexType>
    <xsd:element name="BRAND"        type="dig35:tProductDetails" minOccurs="0"   maxOccurs="1"/>
    <xsd:element ref="CATEGORY"                                   minOccurs="0"   maxOccurs="1"/>
    <xsd:element name="FILM_SIZE"    type="dig35:tDoubleSize"     minOccurs="0"   maxOccurs="1"/>
    <xsd:element name="ROLL_ID"      type="dig35:tLangString"     minOccurs="0"   maxOccurs="1"/>
    <xsd:element name="FRAME_ID"     type="xsd:positiveInteger"   minOccurs="0"   maxOccurs="1"/>
    <xsd:element name="FILM_SPEED"   type="xsd:positiveInteger"   minOccurs="0"   maxOccurs="1"/>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="CATEGORY">
  <xsd:simpleType base="xsd:string">
    <xsd:enumeration value="Negative B/W"/>
    <xsd:enumeration value="Negative Color"/>
    <xsd:enumeration value="Reversal B/W"/>
    <xsd:enumeration value="Reversal Color"/>
    <xsd:enumeration value="Chromagenic"/>
    <xsd:enumeration value="Internegative B/W"/>
    <xsd:enumeration value="Internegative Color"/>
  </xsd:simpleType>
</xsd:element>
```

Brand: This field specifies the name of the film manufacturer. See section F.2.18 for the format of this field.

Category: This field specifies the category of film used. Note that the category `Chromagenic` refers to B/W negative film that is developed with a C41 process (i.e., color negative chemistry).

Film Size: This field specifies the size of the X and Y dimension of the film used, and the unit is in meters.

Roll ID: This field specifies the roll number or ID of the film. For some film, this number is encoded on the film cartridge as a bar code.

Frame ID: This field specifies the frame number or ID of the frame digitized from the roll of film.

Film Speed: This field specifies the film speed of the film. This field is measured in ASA.

```
Example:

<FILM>
  <BRAND>
    <MANUFACTURER>
      <ORG_NAME>Acme</ORG_NAME>
    </MANUFACTURER>
    <MODEL>Aerial 100</MODEL>
  </BRAND>

  <CATEGORY>Reversal B/W</CATEGORY>
  <FILM_SIZE>
    <WIDTH>0.00525</WIDTH>
    <HEIGHT>35</HEIGHT>
  </FILM_SIZE>
  <ROLL_ID>Xr6784YY</ROLL_ID>
  <FRAME_ID>10</FRAME_ID>
  <FILM_SPEED>200</FILM_SPEED>
</FILM>
```

B.4 Examples

B.4.1 Creation Information of a Digital Camera

The following example shows a typical "Image Creation" metadata created by a digital camera. Note that all information may not come directly from a digital camera itself.

```
Example:

<METADATA>
   <IMAGE_CREATION>
      <GENERAL_CREATION_INFO>
         <CREATION_TIME>2000-06-08T21:07:00</CREATION_TIME>
         <IMAGE_SOURCE>Digital Camera</IMAGE_SOURCE>
         <SCENE_TYPE>Original Scene</SCENE_TYPE>
      </GENERAL_CREATION_INFO>
      <CAMERA_CAPTURE>
         <CAMERA_INFO>
            <MANUFACTURER>
               <ORG_NAME>Acme</ORG_NAME>
            </MANUFACTURER>
            <MODEL>MODEL 1000</MODEL>
         </CAMERA_INFO>
         <DEVICE_CHARACTER>
            <MIN_F_NUMBER>2.0</MIN_F_NUMBER>
         </DEVICE_CHARACTER>
         <CAMERA_SETTINGS>
            <R_EXP_TIME>1/60</R_EXP_TIME>
            <F_NUMBER>2.8</F_NUMBER>
            <METERING_MODE>Center Weighted Average</METERING_MODE>
            <FLASH>true</FLASH>
         </CAMERA_SETTINGS>
      </CAMERA_CAPTURE>
   </IMAGE_CREATION>
</METADATA>
```

B.4.2 Creation Information of a Scanner

The following example shows a typical "Image Creation" metadata created by a scanner. Note that all information may not come directly from a scanner itself.

```
Example:

<METADATA>
   <IMAGE_CREATION>
      <GENERAL_CREATION_INFO>
         <CREATION_TIME>2000-06-08T21:07:00</CREATION_TIME>
         <IMAGE_SOURCE>Reflection Print Scanner</IMAGE_SOURCE>
         <SCENE_TYPE>Second Generation Scene</SCENE_TYPE>
      </GENERAL_CREATION_INFO>
      <SCANNER_CAPTURE>
         <SCANNER_INFO>
            <MANUFACTURER>
               <ORG_NAME>Acme</ORG_NAME>
            </MANUFACTURER>
            <MODEL>MODEL 5S</MODEL>
         </SCANNER_INFO>
         <SCANNER_SETTINGS>
            <PIXEL_SIZE>0.000106</PIXEL_SIZE>
            <PHYSICAL_SCAN_RES>
               <WIDTH>0.21</WIDTH>
               <HEIGHT>0.297</HEIGHT>
            </PHYSICAL_SCAN_RES>
         </SCANNER_SETTINGS>
      </SCANNER_CAPTURE>
      <CAPTURED_ITEM>
         <REFLECTION_PRINT>
            <DOCUMENT_SIZE>
               <WIDTH>0.21</WIDTH>
               <HEIGHT>0.297</HEIGHT>
            </DOCUMENT_SIZE>
            <MEDIUM>Line Art</MEDIUM>
            <RP_TYPE>B/W Document</RP_TYPE>
         </REFLECTION_PRINT>
      </CAPTURED_ITEM>
   </IMAGE_CREATION>
</METADATA>
```

Annex C: Content Description Metadata

C.1 Overview

This Annex comprises the *content description* of an image. The content description has two main purposes:

Firstly – it can be used to classify the image. Images placed in a database need to be extracted from that database. For any image to be useful (happy snaps saved in the file system of a personal computer through to an extensive professional photo library), this is required. This classification may be used to search for images.

Secondly – once an image is retrieved, some data that describes the image but is not useful when searching may be included. For example – "Craig is the guy asleep on the lounge" is not all that useful when searching, but is useful when describing the content.

The metadata listed in this Annex contains data for both of the above cases.

All fields listed in this Annex are optional unless otherwise stated. A DIG35-compliant metadata reader or editor must understand all fields unless otherwise stated. Note that a DIG35 metadata editor should not remove fields that are not understood when a DIG metadata file is modified.

C.2 Structure

The following diagram illustrates the logical structure of the Content Description metadata.

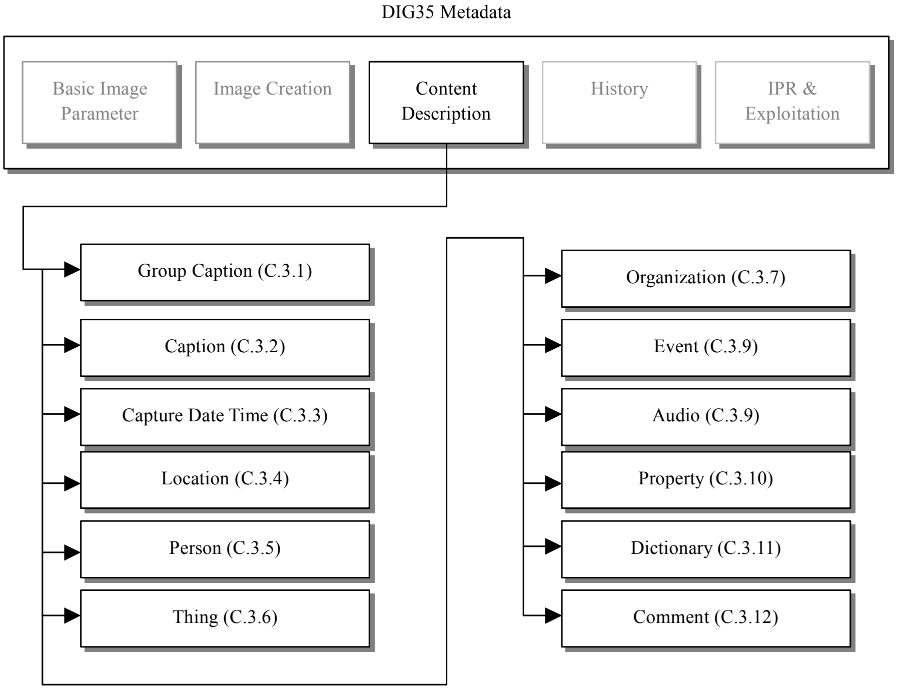


Figure C-1: Content Description metadata structure

C.3 Definition

Content Description Metadata may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="CONTENT_DESCRIPTION">
  <xsd:complexType>
    <xsd:element name="GROUP_CAPTION"type="dig35:tLangString" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="CAPTION"      type="dig35:tLangString" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="CAPTURE_TIME" type="dig35:tDateTime"   minOccurs="0" maxOccurs="1"/>
    <xsd:element name="LOCATION"     type="dig35:tLocation"   minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="dig35:PERSON"                           minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="dig35:THING"                            minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="dig35:ORGANIZATION"                     minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="dig35:EVENT"                            minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="dig35:AUDIO"                            minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="dig35:PROPERTY"                         minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="dig35:DICTIONARY"                       minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="COMMENT"                                minOccurs="0" maxOccurs="1"/>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

Group Caption: This field specifies the subject or purpose of a group. See section C.3.1 for details.

Caption: This field specifies the subject or purpose of the image. See section C.3.2 for details.

Capture Time and Date: This field specifies the time and date the image was initially generated. See section C.3.3 for details.

Location: The section specifies the location of the image. See section C.3.4 for details.

Person Description: This field specifies a person within an image. See section C.3.5 for details.

Thing Description: This field specifies the names of tangible items depicted in the image. See section C.3.6 for details.

Organization Description: This field specifies an organization within an image. See section C.3.7 for details.

Event Description: This field specifies events depicted in the image. See section C.3.8 for details.

Audio: This field specifies audio streams associated with an image. See section C.3.9 for details.

Property: This field specifies information used to describe an image or an object within an image. See section C.3.10 for details.

Dictionary: This field specifies a dictionary of a property. See section C.3.11 for details.

Comment: This field specifies user- and/or application-defined information beyond the scope of other properties in this sub-block. See section C.3.12 for details.

C.3.1 Group Caption

Group Caption specifies the subject or purpose of a group or roll of images (e.g., a roll of film). The image in the digital file is one member of the “roll”.

```
Example:

<GROUP_CAPTION>Bill's Wedding</GROUP_CAPTION>
```

C.3.2 Caption

Caption specifies the subject or purpose of the image. It may be additionally used to provide any other type of information related to the image.

```
Example:

<CAPTION>Bill waiting for the bride</CAPTION>
```

C.3.3 Capture Time and Date

Capture Time and Date specifies the time and date the image was initially generated. This may be different to the capture device date where the capture device is a scanner that scans the image at a different time to when it was initially captured.

See section F.2.8 for the format of this field.

```
Example:

<CAPTURE_TIME>
  <EXACT>2000-01-01T15:10:05</EXACT>
</CAPTURE_TIME>
```

C.3.4 Location

Location specifies the location of the image. This location is the physical location of the image (e.g. address, GPS coordinate), not the position of an object within the image.

See section F.2.15 for the format of this field.

C.3.5 Person Description

Person Description specifies a person within an image. See section F.2.13 for the format of this field. Additionally, this field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="PERSON">
  <xsd:complexType base="dig35:tPerson" derivedBy="extension">
    <xsd:element name="POSITION"     type="dig35:tPosition"   minOccurs="0"   maxOccurs="1"/>
    <xsd:element name="LOCATION"     type="dig35:tLocation"   minOccurs="0"   maxOccurs="1"/>
    <xsd:element ref="dig35:PROPERTY"                         minOccurs="0"   maxOccurs="unbounded"/>
  </xsd:complexType>
</xsd:element>
```

Position: This field specifies the position of the person within the image. See section F.2.17 for the format of this field.

Location: This field specifies the physical location of the person. Note that this field does not specify the relative position of the person. See section F.2.15 for the format of this field.

Property: This field specifies additional information describing the person. See section C.3.10 for the format of this field.

```
Example:

<PERSON ID="1">
  <PERSON_NAME>
    <NAME_COMP TYPE="Given">Bill</NAME_COMP>
    <NAME_COMP TYPE="Family">Billson</NAME_COMP>
  </PERSON_NAME>

  <POSITION>
    <RECT>
      <X>0.2</X>
      <Y>0.2</Y>
      <WIDTH>0.1</WIDTH>
      <HEIGHT>0.4</HEIGHT>
    </RECT>
  </POSITION>
</PERSON>
```

C.3.6 Thing Description

Thing Description specifies the names of tangible things depicted in the image (Washington Monument, for example). This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="THING">
  <xsd:complexType>
    <xsd:element name="NAME"         type="dig35:tLangString" minOccurs="0"   maxOccurs="1"/>
    <xsd:element ref="dig35:COMMENT"                          minOccurs="0"   maxOccurs="1"/>
    <xsd:element name="POSITION"     type="dig35:tPosition"   minOccurs="0"   maxOccurs="1"/>
    <xsd:element name="LOCATION"     type="dig35:tLocation"   minOccurs="0"   maxOccurs="1"/>
    <xsd:element ref="dig35:PROPERTY"                         minOccurs="0"   maxOccurs="unbounded"/>
    <xsd:element ref="THING"                                  minOccurs="0"   maxOccurs="unbounded"/>

    <xsd:attribute name="ID"         type="xsd:string"/>
    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

Name: This field specifies the name of the Thing.

Comment: This field specifies user- and/or application-defined information beyond the scope of other properties in the Thing Description. See section F.4.1 for the format of this field.

Position: This field specifies the position of the thing within the image. See section F.2.17 for the format of this field.

Location: This field specifies the physical location of the thing. Note that this field does not specify the relative position of the thing. See section F.2.15 for the format of this field.

Property: This field specifies additional information describing the thing. See section C.3.10 for the format of this field.

Sub-Thing: This field specifies sub-Things of the encompassing thing.

ID: This field specifies the unique identifier for the thing.

```
Example:

<THING ID="2">
  <NAME>Stonehenge</NAME>

  <THING ID="3">     <!-- Subthing -->
    <NAME>Rock</NAME>
  </THING>
</THING>
```

C.3.7 Organization Description

Organization Description specifies an organization depicted within an image. See section F.2.14 for the format of this field. Additionally, this field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="ORGANIZATION">
  <xsd:complexType base="dig35:tOrganization" derivedBy="extension">
    <xsd:element name="POSITION"     type="dig35:tPosition"    minOccurs="0"   maxOccurs="1"/>
    <xsd:element name="LOCATION"     type="dig35:tLocation"    minOccurs="0"   maxOccurs="1"/>
    <xsd:element ref="dig35:PROPERTY"                          minOccurs="0"   maxOccurs="unbounded"/>
  </xsd:complexType>
</xsd:element>
```

Position: This field specifies the position of the organization within the image. See section F.2.17 for the format of this field.

Location: This field specifies the physical location of the organization. Note that this field does not specify the relative position of the organization. See section F.2.15 for the format of this field.

Property: This field specifies additional information describing the organization. See section C.3.10 for the format of this field.

```
Example:

<ORGANIZATION ID="3">
  <ORG_NAME>Acme Inc</ORG_NAME>

  <LOCATION>
    <COMMENT>Moon surface scanning group headquarters</COMMENT>
  </LOCATION>
</ORGANIZATION>
```

C.3.8 Event Description

Event Description specifies a description of the event depicted in the image. An event is a common reason an image is captured. The event fields in this section address the following requirements:

- Support describing a concrete or abstract event or action
- Define the type of event
- Describe the location of the event
- Support absolute or relative date of the event
- Describe participants and their roles in the event
- Support an association with sub-Events
- Support semantic relationships between events and participants
- Support a collection of event metadata

Examples:

```
Birthday
Anniversary
New Year's Eve
A Coronation
The Crimean War
Hurricane Andrew
```

```
Football Game
      Scrum
         Participants: Bev, Kats, Howard, Jean, Scott, Craig, Rob, Warren
      Kick for Goal
         Participant: Bev
```

Note the event name "Football Game" in the context above refers to a Rugby match. During a rugby match a "Scrum" restarts play after an error by the attacking team.

An example of a Birthday event with Party/Cake/Birthday Song/ Blowing Out Candles Sub-Events:

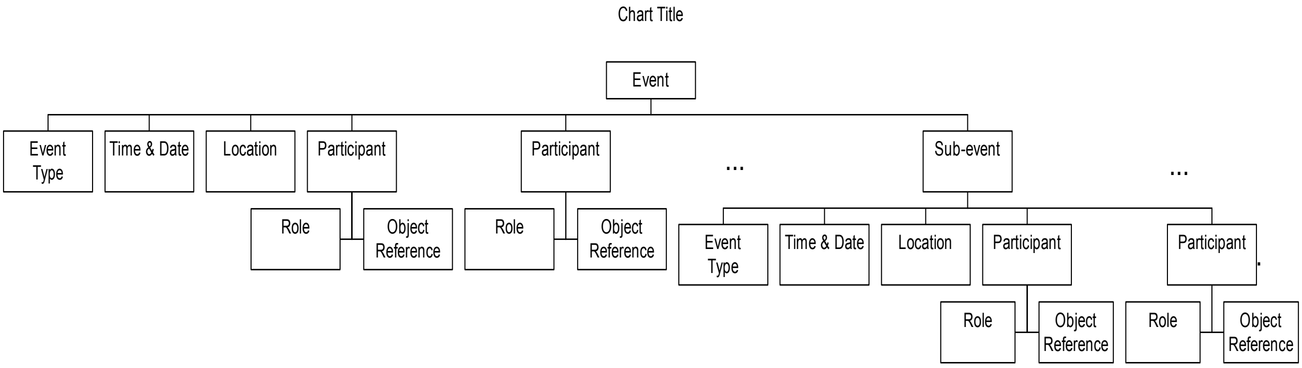
```
Birthday
         Participants: Bev, Kats, Rob, Craig, Warren, Jean
Party Games
         Participants: Bev, Kats, Rob, Craig, Warren, Jean
Cake
         Participants: Bev, Kats, Rob, Craig, Warren, Jean
Birthday Song
         Participants: Kats, Rob, Craig, Warren, Jean
Blowing Out Candles
         Participants: Bev
```

The total event is described by a tree of event metadata. The root may represent an instantaneous event or action (e.g. Anne dances in a competition) with no sub-Events. Alternatively an event of significant duration (e.g., a family reunion) may include sub-Events. The sub-Events might represent the various phases of an event (e.g. arrivals, games, the picnic dinner, after-dinner conversations, and departures for the family reunion) and include type, time, location and participants. A leaf level of an event tree is an event with no sub-Events. A representation of a leaf level event is a specific instant (e.g. a particularly impressive spike in the volleyball game before the dinner at the reunion).

Each event field has sub-fields to define or describe:

- The event or action type.
- The time of the event
- The duration of the event
- The description of the event
- The location of the event
- The participants in the event
- The roles of the participants
- A comment field that describes any additional event information
- The sub-events of the event

The following diagram shows an example structure of the event fields, Duration Description and Comment fields were not included:



This field may contain the sub-fields listed below unless otherwise stated.

```
Schema Definition:

<xsd:element name="EVENT">
  <xsd:complexType>
    <xsd:element name="EVENT_TYPE"   type="dig35:tLangString"/>
    <xsd:element name="DESCRIPTION"  type="dig35:tLangString" minOccurs="0"  maxOccurs="1"/>

    <xsd:element name="LOCATION"     type="dig35:tLocation"   minOccurs="0"  maxOccurs="1"/>
    <xsd:element name="EVENT_TIME"   type="dig35:tDateTime"   minOccurs="0"  maxOccurs="1"/>
    <xsd:element name="DURATION"     type="xsd:timeDuration"  minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="dig35:COMMENT"                          minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="PARTICIPANT"                            minOccurs="0"  maxOccurs="unbounded"/>
    <xsd:element ref="EVENT_RELATION"                         minOccurs="0"  maxOccurs="unbounded"/>
    <xsd:choice                                               minOccurs="0"  maxOccurs="unbounded">
      <xsd:element ref="EVENT"/>
      <xsd:element name="EVENT_REF"  type="xsd:string"/>
    </xsd:choice>

    <xsd:attribute name="ID"         type="xsd:string"/>
    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="PARTICIPANT">
  <xsd:complexType>
    <xsd:element name="ROLE"         type="dig35:tLangString"   minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="OBJECT_REF"   type="xsd:string"/>

    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="EVENT_RELATION">
  <xsd:complexType>
    <xsd:element name="RELATION"     type="dig35:tLangString"  minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="EVENT_REF"    type="xsd:string"                       maxOccurs="unbounded"/>
  </xsd:complexType>
</xsd:element>
```

ID: This field specifies the unique identifier for the Event.

Event Type: This field shall occur within the Event Description field. The field specifies the type of the Event.

Description: This field specifies the description of the Event. This field is used to describe an event in human readable text format.

Location: This field identifies the physical location of the Event and not the position within the image. See section F.2.15 for the format of this field.

Event Date and Time: This field specifies the start time of the event. See section F.2.8 for the format of this field.

Duration: This field specifies the duration of the Event.

Comment: This field specifies user- and/or application-defined information beyond the scope of other properties in the Event Description field. See section F.4.1 for the format of this field.

Participant: This field specifies the participants of the Event. A participant may be a person, an organization or a thing.

Role: This field specifies the role of the participant within the Event.

Object Reference: This field is a reference to a participant. This field is a link to one of the Person Description, Organization Description or Thing Description fields.

Event Relationships: This field specifies relationships to other Events. These are used for relationships between Events that are not directly sub-Events of each other. An example of a relationship might be a link to a previous Event of the same type.

Relation: This field specifies a description of the relationship(s) to the other Event(s).

Event Reference: This field is a reference to the related Events. This field is a link to one of the other Event Description fields.

Sub-Events: This field specifies sub-Events of the encompassing event. A sub-Event field may contain sub-Events. The sub-Event field may be either contained within the Event field, or referenced:

Event: This field specifies the sub-Event.

Event Reference: This field is a reference to the sub-Event. This field is a link to one of the other Event Description fields.

Note that a sub-Event may be included either directly, or by reference. Within a collection, this allows an event that describes the entire collection to contain a sub-event that describes only some of the images in the collection.

```
Example 1:

<!--
   - The following example is of the rugby scrum described above
  -->

<!--
   - Person descriptions
  -->
<PERSON ID="1001">
  <PERSON_NAME>
    <NAME_COMP TYPE="Given">Bev</NAME_COMP>
  </PERSON_NAME>
</PERSON>
<PERSON ID="1002">
  <PERSON_NAME>
    <NAME_COMP TYPE="Given">Kats</NAME_COMP>
  </PERSON_NAME>
</PERSON>
<PERSON ID="1003">
  <PERSON_NAME>
    <NAME_COMP TYPE="Given">Howard</NAME_COMP>
  </PERSON_NAME>
</PERSON>
<PERSON ID="1004">
  <PERSON_NAME>
    <NAME_COMP TYPE="Given">Jean</NAME_COMP>
  </PERSON_NAME>
</PERSON>
<PERSON ID="1005">
  <PERSON_NAME>
    <NAME_COMP TYPE="Given">Scott</NAME_COMP>
  </PERSON_NAME>
</PERSON>
<PERSON ID="1006">
  <PERSON_NAME>
    <NAME_COMP TYPE="Given">Craig</NAME_COMP>
  </PERSON_NAME>
</PERSON>
<PERSON ID="1007">
  <PERSON_NAME>
    <NAME_COMP TYPE="Given">Rob</NAME_COMP>
  </PERSON_NAME>
</PERSON>
<PERSON ID="1008">
  <PERSON_NAME>
```

```
    <NAME_COMP TYPE="Given">Warren</NAME_COMP>
  </PERSON_NAME>
</PERSON>

<!--
   - Event descriptions
  -->
<EVENT>
  <EVENT_TYPE>Football Game</EVENT_TYPE>
  <LOCATION>
    <ADDRESS>
      <COUNTRY>Australia</COUNTRY>
    </ADDRESS>
  </LOCATION>
  <COMMENT>A game of Rugby.  An international game similar to American Football.</COMMENT>

  <EVENT>
    <EVENT_TYPE>Scrum</EVENT_TYPE>
    <PARTICIPANT>
      <ROLE>Loosehead Prop</ROLE>
      <OBJECT_REF>1001</OBJECT_REF>
    </PARTICIPANT>
    <PARTICIPANT>
      <ROLE>Hooker</ROLE>
      <OBJECT_REF>1002</OBJECT_REF>
    </PARTICIPANT>
    <PARTICIPANT>
      <ROLE>Tighthead Prop</ROLE>
      <OBJECT_REF>1003</OBJECT_REF>
    </PARTICIPANT>
    <PARTICIPANT>
      <ROLE>Left Lock</ROLE>
      <OBJECT_REF>1004</OBJECT_REF>
    </PARTICIPANT>
    <PARTICIPANT>
      <ROLE>Right Lock</ROLE>
      <OBJECT_REF>1005</OBJECT_REF>
    </PARTICIPANT>
    <PARTICIPANT>
      <ROLE>Blindside Flanker</ROLE>
      <OBJECT_REF>1006</OBJECT_REF>
    </PARTICIPANT>
    <PARTICIPANT>
      <ROLE>Openside Flanker</ROLE>
      <OBJECT_REF>1007</OBJECT_REF>
    </PARTICIPANT>
    <PARTICIPANT>
      <ROLE>Breakaway</ROLE>
      <OBJECT_REF>1008</OBJECT_REF>
    </PARTICIPANT>
  </EVENT>

  <EVENT>
    <EVENT_TYPE>Kick for Goal</EVENT_TYPE>
    <PARTICIPANT>
      <ROLE>Kicker</ROLE>
      <OBJECT_REF>1001</OBJECT_REF>
    </PARTICIPANT>
  </EVENT>

</EVENT>
```

Example 2:

```
<!--
   - The following example of an Event XML instance metadata for am image that
   -  contains the following:
   - The picture was captured at a Birthday Party
   - The following three people are in the image:
      - Morgan the birthday girl
      - Melissa the guest who is Morgan’s best friend
      - Debora the host who is Morgan's mother
   -->

<!--
   - Person descriptions
  -->
<PERSON ID="1001">
  <PERSON_NAME>
    <NAME_COMP TYPE="Given">Morgan</NAME_COMP>
  </PERSON_NAME>
  <BIRTH_DATE>1989-08-15</BIRTH_DATE>
</PERSON>
<PERSON ID="1002">
  <PERSON_NAME>
    <NAME_COMP TYPE="Given">Melissa</NAME_COMP>
  </PERSON_NAME>
  <COMMENT>Morgan’s best friend</COMMENT>
</PERSON>
<PERSON ID="1003">
  <PERSON_NAME>
    <NAME_COMP TYPE="Given">Debora</NAME_COMP>
  </PERSON_NAME>
  <COMMENT>Morgan’s mother</COMMENT>
</PERSON>

<!--
   - Event descriptions
  -->
<EVENT ID="0100">
  <EVENT_TYPE>Birthday</EVENT_TYPE>
  <EVENT_TIME>
    <EXACT>2000-08-15T13:20:00.000-05:00</EXACT>
  </EVENT_TIME>
  <!-- August 15 2000 at 1:20pm Eastern Standard Time which is 5 hours behind
       Coordinated Universal Time-->
  <DURATION>PT3H</DURATION>  <!-- Duration of birthday party was 3 hours -->

  <PARTICIPANT>
    <ROLE>Birthday Girl</ROLE>
    <OBJECT_REF>1001</OBJECT_REF>
  </PARTICIPANT>
  <PARTICIPANT>
    <ROLE>Guest</ROLE>
    <OBJECT_REF>1002</OBJECT_REF>
  </PARTICIPANT>
  <PARTICIPANT>
    <ROLE>Host</ROLE>
    <OBJECT_REF>1003</OBJECT_REF>
  </PARTICIPANT>
</EVENT>
```

C.3.9 Audio

Audio specifies an audio stream associated with the image. This field may contain the sub-fields listed below unless otherwise specified.

```
Schema Definition:

<xsd:element name="AUDIO">
  <xsd:complexType>
    <xsd:element name="AUDIO_STREAM"  type="xsd:uriReference"/>
    <xsd:element name="AUDIO_FORMAT"  type="xsd:string"         minOccurs="0"  maxOccurs="1"/>
    <xsd:element name="MIME_TYPE"     type="xsd:string"         minOccurs="0"  maxOccurs="1"/>
    <xsd:element name="DESCRIPTION"   type="dig35:tLangString"  minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="COMMENT"                                  minOccurs="0"  maxOccurs="1"/>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

Audio Stream: This field shall occur within the Audio field. This field specifies a reference to an audio stream.

Audio Stream Format: This field specifies the name of the audio stream format.

Table C-1: Suggested Audio Stream Format values

Values	Meaning
AIFF	Audio Interchange File Format
MIDI	Music Instrument Digital Interface file format
MP3	MPEG audio layer 3
WAV	Wave Form Audio File Format

MIME Type: This field specifies the Internet media type of the audio file.

Description: This field specifies a description of the audio stream.

Comment: This field specifies user- and/or application-defined information beyond the scope of other properties in the audio. See section F.4.1 for the format of this field.

```
Example:

<AUDIO>
  <AUDIO_STREAM>0001.mp3</AUDIO_STREAM>
  <AUDIO_FORMAT>MP3</AUDIO_FORMAT>
  <DESCRIPTION>Description of the image.</DESCRIPTION>
</AUDIO>
```

C.3.10 Property

Property specifies a description an image or an object within an image. This field shall contain a name and may contain a value and sub-property fields. A Property is either a single word or a small phrase and a value. The Property is a non-exact language-specific definition of the image or part of the image. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="PROPERTY">
  <xsd:complexType>
    <xsd:element name="NAME"          type="dig35:tLangString" minOccurs="0"   maxOccurs="1"/>
    <xsd:element name="VALUE"         type="dig35:tLangString" minOccurs="0"   maxOccurs="unbounded"/>
    <xsd:element ref="dig35:COMMENT"                           minOccurs="0"   maxOccurs="1"/>
    <xsd:element ref="PROPERTY"                                minOccurs="0"   maxOccurs="unbounded"/>

    <xsd:attribute name="DICT_REF"    type="xsd:string"/>
    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

Name: This field specifies the name of the Property.

Value: This field specifies the property value. A Property that contains a value shall not contain sub-Property fields.

Comment: This field specifies user- and/or application-defined information beyond the scope identified in the Property. See section F.4.1 for the format of this field.

Sub-Property: This field specifies sub-Properties of the encompassing Property. A property that contains sub-Property fields shall not contain a value.

Dictionary Reference: This field specifies a reference to a Dictionary. See C.3.11 for more detail.

```
Example:

<PROPERTY DICT_REF="10">
  <NAME>Quality</NAME>
  <VALUE>Good</VALUE>
</PROPERTY>

<PROPERTY>
  <NAME>Weight</NAME>
  <VALUE>75kg</VALUE>
</PROPERTY>

<PROPERTY>
  <NAME>Texture</NAME>
  <VALUE>Red</VALUE>
  <VALUE>Yellow</VALUE>
  <VALUE>Rough</VALUE>
</PROPERTY>

<PROPERTY>
  <NAME>Metadata Standard</NAME>
  <PROPERTY>
    <NAME>Specification</NAME>
    <PROPERTY>
      <NAME>Figure</NAME>
      <VALUE>12B</VALUE>
    </PROPERTY>
  </PROPERTY>
</PROPERTY>
```

C.3.11 Dictionary Definition

Dictionary Definition specifies the name of a dictionary. A Property may be defined using a specific dictionary. The advantage of this is that there is a single definition for each Property, and that two different Property annotations are not used to define the same thing.

To give an example, a dictionary may define the word "Vehicle" to be used to describe a car, bus, taxi, truck, automobile, etc. A second example is the use of the word "Date". Date could be used to specify the fruit of the palm "date" and not the definition of date as a day.

This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="DICTIONARY">
  <xsd:complexType>
    <xsd:element name="DICT_NAME"    type="dig35:tLangString" minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="dig35:COMMENT"                          minOccurs="0"  maxOccurs="1"/>

    <xsd:attribute name="ID"         type="xsd:string"/>
    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

Dictionary Name: This field specifies the name of the dictionary used to define a Property.

Comment: This field specifies user- and/or application-defined information related to the Dictionary. See section F.4.1 for the format of this field.

Dictionary ID: This field specifies the unique identifier for the dictionary.

```
Example:

<DICTIONARY ID="101">
  <!--
     - Dictionary from The Getty Vocabulary Program. © 2000 J. Paul Getty Trust
    -->
  <DICT_NAME>The Art and Architecture Thesaurus</DICT_NAME>
</DICTIONARY>
```

C.3.12 Comment

Comment contains user- and/or application-defined information beyond the scope defined in this Annex.

See section F.4.1 for the format of this field.

C.4 Examples

```
<METADATA>
   <CONTENT_DESCRIPTION>
      <GROUP_CAPTION>Bill's Wedding</GROUP_CAPTION>
      <CAPTION>Bill waiting for the bride</CAPTION>
      <CAPTURE_TIME>
         <EXACT>2000-01-01T15:10:05</EXACT>
      </CAPTURE_TIME>
      <PERSON ID="1">
         <PERSON_NAME>
            <NAME_COMP TYPE="Given">Bill</NAME_COMP>
            <NAME_COMP TYPE="Family">Billson</NAME_COMP>
         </PERSON_NAME>
         <POSITION>
            <RECT>
               <X>0.2</X>
               <Y>0.2</Y>
               <WIDTH>0.1</WIDTH>
               <HEIGHT>0.4</HEIGHT>
            </RECT>
         </POSITION>
      </PERSON>
   </CONTENT_DESCRIPTION>
</METADATA>
```

Annex D: History Metadata

D.1 Overview

This Annex comprises the *history* of the metadata of an image. The History metadata is used to provide *partial information* about how the image got to the present state. This metadata is only approximate because;

- some of the data (both metadata and image data) is collapsed, thus providing only a summary
- some of the data may not have been properly entered because applications used were not capable of updating the history metadata.

The History metadata contains a summary of basic image editing operations that have already been applied to the image and previous version(s) of the image metadata. Note that the History metadata is not designed to be used to reverse ("undo") image editing operations.

D.2 Structure

The following diagram illustrates the logical structure of the History metadata. Dotted lines indicate that there may be optional sub-fields present in the History metadata structure.

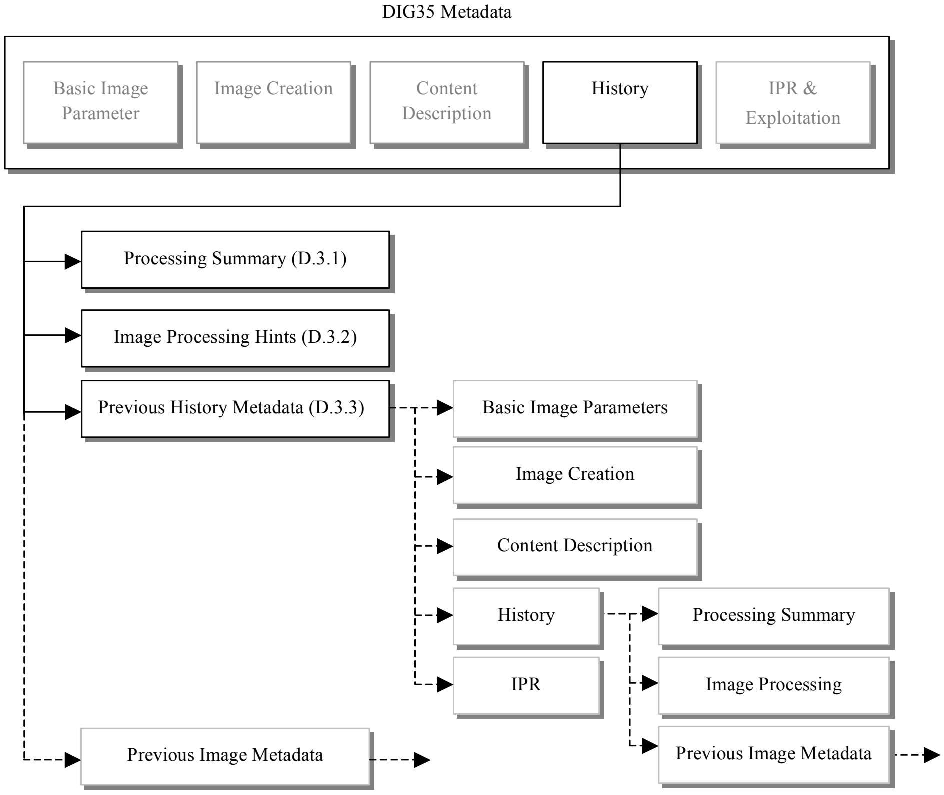


Figure D-1: History metadata structure

D.3 Definition

History Metadata may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="HISTORY">
  <xsd:complexType>
    <xsd:element ref="dig35:PROCESSING_SUMMARY"             minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="dig35:IMAGE_PROCESSING_HINTS"         minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="dig35:METADATA"                       minOccurs="0"  maxOccurs="unbounded"/>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

Processing Summary: This field specifies a list of operations previously applied to an image during the course of its workflow. See section D.3.1 for details.

Image Processing Hints: This field specifies a list of the operations previously performed when editing an image. See section D.3.2 for details.

Previous History Metadata: This field specifies a previous version of the metadata that may include metadata about portions of an image that has deleted (cropped for example). This field is also useful when multiple images are combined to create a new image. See section D.3.3 for details.

D.3.1 Processing Summary

Processing Summary specifies a list of the operations performed over the life of the image listing the operations performed and not the ordering or the number of times each operation is performed.

The Processing Summary defined below should be considered as potential and in all likelihood partial information. That is because the presence of a particular hint, such as “Image Cropped,” indicates that the image has been cropped. However, absence of a “Image Cropped” hint is no assurance that the image has never been cropped.

This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="PROCESSING_SUMMARY">
  <xsd:complexType>
    <xsd:element name="IMG_CREATED"                         minOccurs="0"  maxOccurs="1">
      <xsd:complexType content="empty"/>
    </xsd:element>
    <xsd:element name="IMG_CROPPED"                         minOccurs="0"  maxOccurs="1">
      <xsd:complexType content="empty"/>
    </xsd:element>
    <xsd:element name="IMG_TRANSFORMED"                     minOccurs="0"  maxOccurs="1">
      <xsd:complexType content="empty"/>
    </xsd:element>
    <xsd:element name="IMG_GTC_ADJ"                         minOccurs="0"  maxOccurs="1">
      <xsd:complexType content="empty"/>
    </xsd:element>
    <xsd:element name="IMG_STC_ADJ"                         minOccurs="0"  maxOccurs="1">
      <xsd:complexType content="empty"/>
    </xsd:element>
    <xsd:element name="IMG_SPATIAL_ADJ"                     minOccurs="0"  maxOccurs="1">
      <xsd:complexType content="empty"/>
    </xsd:element>
    <xsd:element name="IMG_EXT_EDITED"                      minOccurs="0"  maxOccurs="1">
      <xsd:complexType content="empty"/>
    </xsd:element>
    <xsd:element name="IMG_RETOUCHED"                       minOccurs="0"  maxOccurs="1">
```

```
      <xsd:complexType content="empty"/>
    </xsd:element>
    <xsd:element name="IMG_COMPOSITED"                       minOccurs="0"  maxOccurs="1">
      <xsd:complexType content="empty"/>
    </xsd:element>
    <xsd:element name="IMG_METADATA"                         minOccurs="0"  maxOccurs="1">
      <xsd:complexType content="empty"/>
    </xsd:element>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
  </xsd:complexType>
</xsd:element>
```

Digital Image Created: The presence of this field indicates that the image was created by a metadata-aware application or process.

Note that where a number of operations are performed in the creation of an image (such as removing borders), they should be summarized using the Digital Image Created operation and not listed independently.

This field is especially useful to show truncation of image metadata. Where this field is not present, the full history of the metadata is known to be incomplete. Presence of this field does not indicate that the metadata history is complete though.

Image Cropped: The presence of this field indicates that an image editing application, program, or system has cropped the image.

Image Transformed: The presence of this field indicates that an image has been transformed.

Global Tone / Color Adjustment: The presence of this field indicates that a contrast or density adjustment has been applied to the image, or that the image coloring has been adjusted.

Selective Tone / Color Adjustment: The presence of this field indicates that a contrast or density adjustment has been applied to a selected region of the image.

Global Spatial Adjustment: The presence of this field indicates that the image has been sharpened, or compressed, or blurred, or re-sampled.

Pixels Extensively Edited: The presence of this field indicates the image has been edited extensively – enough to change the captured scene content.

Image Retouched: The presence of this field indicates the image pixels have been edited to remove scratches or red-eye, or other minor image blemishes.

Image Composited: The presence of this field indicates the image has been created by compositing an image with another image, or a background, graphic, or text.

Metadata Adjusted: The presence of this field indicates the image metadata has been modified.

```
Example:
<PROCESSING_SUMMARY>
  <IMG_CREATED/>
  <IMG_METADATA/>
</PROCESSING_SUMMARY>
```

D.3.2 Image Processing Hints

Image Processing Hints specifies a list of the operations performed when editing an image. They differ from the Processing Summary in that the hints list all the operations in order and the operations may be listed more than once (if the operation was used more than once).

The Processing Summary lists a subset of all the operations performed during the life of an image while the Image Processing Hints field stores the most current set of operations in greater detail.

The complete list of operations (and their order) can be generated by combining all Image Processing Hints within a Metadata History tree.

The Image Processing Hints field contains the same fields as the Processing Summary field. Note that each field may appear more than once within each field and each field may contain a textual description of the operation. The Image Processing Hints defined below should be considered potentially partial information. That is because the presence of a particular hint, such as “Image Cropped,” indicates that the image has been cropped and metadata may have been omitted at the same time. However, absence of a “Image Cropped” hint is no assurance that the image has never been cropped.

This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="IMAGE_PROCESSING_HINTS">
  <xsd:complexType>
    <xsd:choice                                                  minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="IMG_CREATED"      type="dig35:tLangString"/>
      <xsd:element name="IMG_CROPPED"      type="dig35:tLangString"/>
      <xsd:element name="IMG_TRANSFORMED"  type="dig35:tLangString"/>
      <xsd:element name="IMG_GTC_ADJ"      type="dig35:tLangString"/>
      <xsd:element name="IMG_STC_ADJ"      type="dig35:tLangString"/>
      <xsd:element name="IMG_SPATIAL_ADJ"  type="dig35:tLangString"/>
      <xsd:element name="IMG_EXT_EDITED"   type="dig35:tLangString"/>
      <xsd:element name="IMG_RETOUCHED"    type="dig35:tLangString"/>
      <xsd:element name="IMG_COMPOSITED"   type="dig35:tLangString"/>
      <xsd:element name="IMG_METADATA"     type="dig35:tLangString"/>
    </xsd:choice>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

```
Example:

<IMAGE_PROCESSING_HINTS>
  <IMG_CREATED>Image created using TWAIN transfer from digital camera.</IMG_CREATED>
  <IMG_CROPPED>Train Removed.</IMG_CROPPED>
  <IMG_TRANSFORMED>Rotated by 10 degrees.</IMG_TRANSFORMED>
  <IMG_TRANSFORMED>Flipped vertically.</IMG_TRANSFORMED>
  <IMG_GTC_ADJ>Image contrast increased.</IMG_GTC_ADJ>
  <IMG_STC_ADJ>Red-eye removed.</IMG_STC_ADJ>
  <IMG_SPATIAL_ADJ>Image sharpened.</IMG_SPATIAL_ADJ>
  <IMG_EXT_EDITED>Modified to after plastic surgery.</IMG_EXT_EDITED>
  <IMG_RETOUCHED>Wrinkles removed.</IMG_RETOUCHED>
  <IMG_COMPOSITED>Car composited over background.</IMG_COMPOSITED>
  <IMG_COMPOSITED>Tree composited in a mask.</IMG_COMPOSITED>
  <IMG_METADATA>Address information updated.</IMG_METADATA>
</IMAGE_PROCESSING_HINTS>
```

D.3.3 Previous History Metadata

Previous History Metadata specifies a previous version of the metadata (including previous history information). See Annex G: or the format of this field.

Each time a new image is created as a result of editing an image or combining several images, some of the metadata from the previous image(s) may be moved to or referenced by the image metadata history. The contributing image(s) Basic Image Parameter, Image Creation, Content Description, and IPR metadata may be recorded in a Previous History Metadata field. Careful consideration shall be made with regards to this previous metadata, particularly previous IPR information. If the IPR metadata is independent of other metadata (e.g., contained within a single XML instance document), then all metadata blocks may be included here. However, if the IPR metadata block is separately associated with the image data, then the previous IPR metadata must be contained within the IPR history field to maintain the IPR self-contained status.

D.4 Examples

D.4.1 Simple Image Editing Example

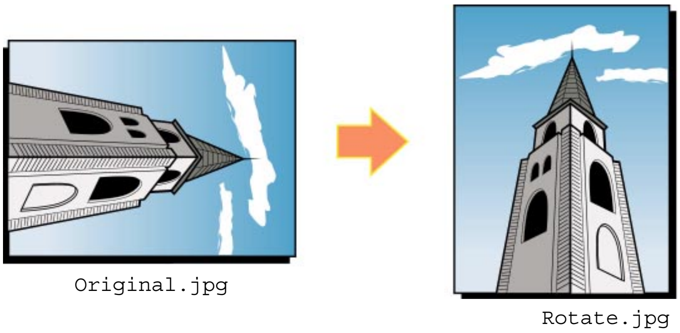


Figure D-2: Simple editing example

Example:

```
<!--
   - Image Original.jpg is created
   - Image Original.jpg has operation (rotation) performed to create image Rotate.jpg
  -->

<!--
   - Metadata for Rotate.jpg
  -->
<METADATA>
  <BASIC_IMAGE_PARAM>
    <BASIC_IMAGE_INFO>
      <FILE_FORMAT>
        <FILE_NAME>Rotate.jpg</FILE_NAME>
      </FILE_FORMAT>
    </BASIC_IMAGE_INFO>
  </BASIC_IMAGE_PARAM>

  <!--
     - Metadata history of Rotate.jpg
    -->
  <HISTORY>
    <PROCESSING_SUMMARY>
       <IMG_CREATED/>
       <IMG_CROPPED/>
       <IMG_TRANSFORMED/>
    </PROCESSING_SUMMARY>
    <IMAGE_PROCESSING_HINTS>
       <IMG_TRANSFORMED>Rotated by 90 degrees.</IMG_TRANSFORMED>
    </IMAGE_PROCESSING_HINTS>

    <!--
       - Previous metadata; this is from Original.jpg
      -->
    <METADATA>
       <BASIC_IMAGE_PARAM>
          <BASIC_IMAGE_INFO>
             <FILE_FORMAT>
                <FILE_NAME>Original.jpg</FILE_NAME>
             </FILE_FORMAT>
          </BASIC_IMAGE_INFO>
       </BASIC_IMAGE_PARAM>
```

```
        <HISTORY>
           <PROCESSING_SUMMARY>
             <IMG_CREATED/>
           </PROCESSING_SUMMARY>
           <IMAGE_PROCESSING_HINTS>
              <IMG_CREATED>Image created using TWAIN transfer from digital camera</IMG_CREATED>
           </IMAGE_PROCESSING_HINTS>
        </HISTORY>
     </METADATA>
   </HISTORY>
</METADATA>
```

D.4.2 Complex Image Editing Example

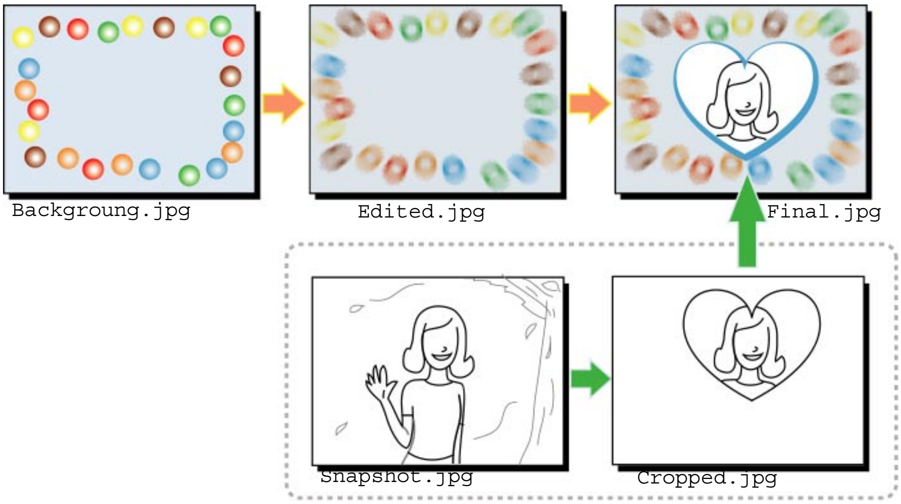


Figure D-3: Complex editing example

```
Example:

<!--
   -  * Image Background.jpg is created
   -  * Image Background.jpg has operation (1),(2) performed to create Rotate.jpg
   -  * Image Snapshot.jpg is created
   -  * Image Snapshot.jpg has been cropped to create Cropped.jpg
   -  * Edited.jpg and Snapshot.jpg have been composited to create Final.jpg
  -->
<METADATA>
  <BASIC_IMAGE_PARAM>
    <BASIC_IMAGE_INFO>
      <FILE_FORMAT>
        <FILE_NAME>Final.jpg</FILE_NAME>
        <FORMAT_TYPE>JFIF</FORMAT_TYPE>
      </FILE_FORMAT>
    </BASIC_IMAGE_INFO>
  </BASIC_IMAGE_PARAM>

  <!--
     - Metadata history of Final.jpg
    -->
  <HISTORY>
  <!--
     - Processing summary on how Final.jpg got to this stage
    -->
    <PROCESSING_SUMMARY>
      <IMG_CREATED/>
      <IMG_CROPPED/>
      <IMG_TRANSFORMED/>
      <IMG_RETOUCHED/>
      <IMG_COMPOSITED/>
    </PROCESSING_SUMMARY>
    <IMAGE_PROCESSING_HINTS>
      <IMG_COMPOSITED>Edited.jpg and Cropped.jpg composited</IMG_COMPOSITED>
    </IMAGE_PROCESSING_HINTS>

    <!--
       - This was the metadata of Edited.jpg that is the base image to create Final.jpg
      -->
    <METADATA>
      <BASIC_IMAGE_PARAM>
        <BASIC_IMAGE_INFO>
          <FILE_FORMAT>
            <FILE_NAME>Edited.jpg</FILE_NAME>
            <FORMAT_TYPE>JPEG</FORMAT_TYPE>
          </FILE_FORMAT>
        </BASIC_IMAGE_INFO>
      </BASIC_IMAGE_PARAM>

      <!--
         - This is the metadata history of Edited.jpg
        -->
      <HISTORY>
        <PROCESSING_SUMMARY>
          <IMG_CREATED/>
          <IMG_TRANSFORMED/>
          <IMG_RETOUCHED/>
        </PROCESSING_SUMMARY>
        <IMAGE_PROCESSING_HINTS>
          <IMG_TRANSFORMED>Operation (1)</IMG_TRANSFORMED>
          <IMG_RETOUCHED>Operation (2)</IMG_RETOUCHED>
        </IMAGE_PROCESSING_HINTS>
        <METADATA>
          <BASIC_IMAGE_PARAM>
            <BASIC_IMAGE_INFO>
              <FILE_FORMAT>
                <FILE_NAME>Background.jpg</FILE_NAME>
                <FORMAT_TYPE>JPEG</FORMAT_TYPE>
              </FILE_FORMAT>
            </BASIC_IMAGE_INFO>
          </BASIC_IMAGE_PARAM>
          <HISTORY>
            <PROCESSING_SUMMARY>
              <IMG_CREATED/>
            </PROCESSING_SUMMARY>
            <IMAGE_PROCESSING_HINTS>
              <IMG_CREATED>Bacground created</IMG_CREATED>
            </IMAGE_PROCESSING_HINTS>
          </HISTORY>
        </METADATA>
      </HISTORY>
    </METADATA>

    <!--
       - This was the metadata of Cropped.jpg
      -->
    <METADATA>
      <BASIC_IMAGE_PARAM>
        <BASIC_IMAGE_INFO>
          <FILE_FORMAT>
```

```
            <FILE_NAME>Cropped.jpg</FILE_NAME>
            <FORMAT_TYPE>JPEG</FORMAT_TYPE>
          </FILE_FORMAT>
        </BASIC_IMAGE_INFO>
      </BASIC_IMAGE_PARAM>
      <HISTORY>
        <PROCESSING_SUMMARY>
          <IMG_CREATED/>
          <IMG_CROPPED/>
        </PROCESSING_SUMMARY>
        <IMAGE_PROCESSING_HINTS>
          <IMG_CROPPED>Cropping has been done.</IMG_CROPPED>
        </IMAGE_PROCESSING_HINTS>
        <METADATA>
          <BASIC_IMAGE_PARAM>
            <BASIC_IMAGE_INFO>
              <FILE_FORMAT>
                <FILE_NAME>Snapshot.jpg</FILE_NAME>
                <FORMAT_TYPE>JPEG</FORMAT_TYPE>
              </FILE_FORMAT>
            </BASIC_IMAGE_INFO>
          </BASIC_IMAGE_PARAM>
          <HISTORY>
            <PROCESSING_SUMMARY>
              <IMG_CREATED/>
            </PROCESSING_SUMMARY>
            <IMAGE_PROCESSING_HINTS>
              <IMG_CREATED>Snapshot was taken</IMG_CREATED>
            </IMAGE_PROCESSING_HINTS>
          </HISTORY>
        </METADATA>
      </HISTORY>
    </METADATA>
  </HISTORY>
</METADATA>
```

Annex E: Intellectual Property Rights Metadata

E.1 Overview

This Annex defines metadata that are related to *Intellectual Property Rights (IPR)* of an image. IPR metadata are designed to protect the contents of an image file against misuse and must preserve both moral rights and copyrights. Moral rights are those rights attached to the creation process; therefore, moral rights persistently pertain to the author or creator of the artwork, whereas copyrights can be repeatedly transferred to different owners, under exploitation conditions that are also part of the IPR and exploitation metadata.

Additional information may be required to maintain IPR and exploitation conditions, such as limitations attached to the purchasing or licensing of copyrights. Names, content description, dates, etc. may also be useful to establish the priority of IPR if different vendors claim IPR to an image.

To ease the processing of IPR-related administrative tasks, identification (e.g., a unique inventory number) and contact point for exploitation are also considered useful metadata. One type of unique identifier is the licence plate as defined in the ISO-JPEG standard, delivered by a Trusted Third Party called a “Registration Authority.” Other fields may be added according to the specificity of the application.

IPR metadata are used to secure the payment of licensing fees and royalties associated with the represented image. Because of international agreements (e.g., the WIPO Treaty), this particular part of the metadata set is subject to specific conditions in terms of persistence. According to the WIPO Treaty, it is strictly forbidden to "remove” or “alter" IPR metadata that has been inserted inside a file; therefore, the only allowable operation is the *addition* of IPR metadata. The IPR History metadata stores all IPR metadata-related modifications. IPR modifications may also be stored as part of the Image History metadata. Maintaining this history is important; in the case that IPR metadata has been removed from a file, the Image History may maintain a record of the intellectual property rights attached to the image, and the author can still claim intellectual property rights.

E.2 Structure

The following diagram illustrates the logical structure of the IPR metadata.

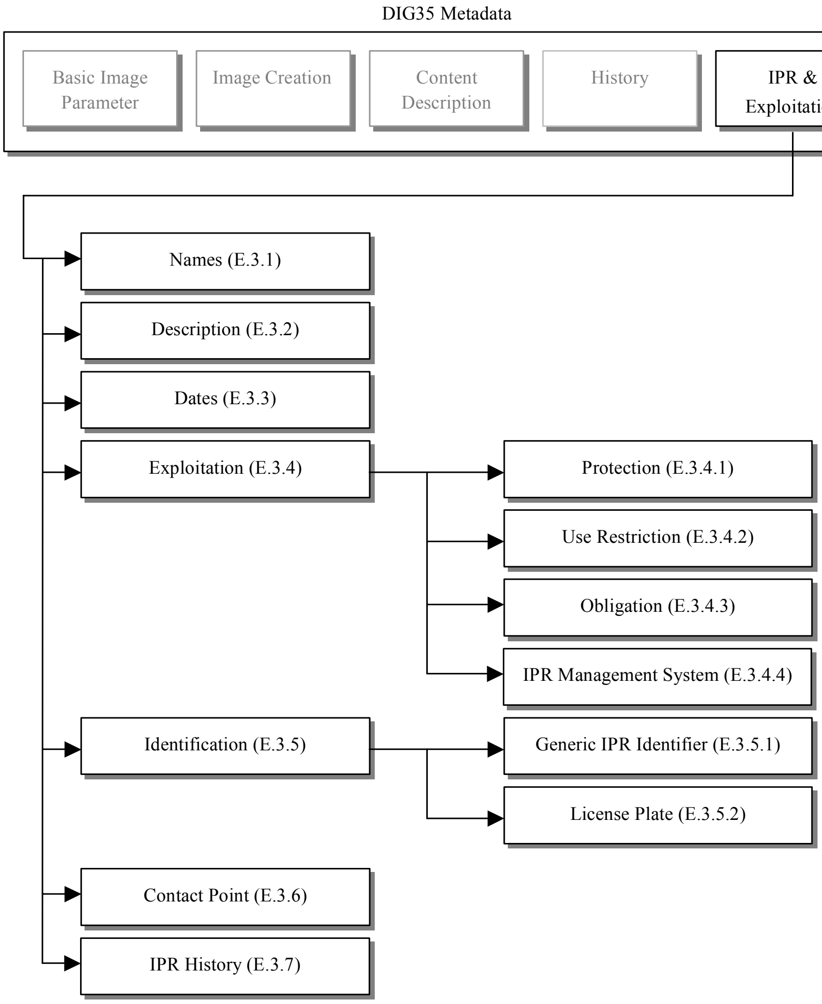


Figure E-1: IPR metadata structure

E.3 Definition

Intellectual Property Rights Metadata may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="IPR">
  <xsd:complexType>
    <xsd:element ref="dig35:IPR_NAMES"                         minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="dig35:IPR_DESCRIPTION"                   minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="dig35:IPR_DATES"                         minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="dig35:IPR_EXPLOITATION"                  minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="dig35:IPR_IDENTIFICATION"                minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="dig35:IPR_CONTACT_POINT"                 minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="dig35:IPR_HISTORY"                       minOccurs="0"  maxOccurs="1"/>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

Names: This field specifies names related to the represented image. See section E.3.1 for details.

Description: This field specifies the description of the content. See section E.3.2 for details.

Dates: This field specifies the IPR-related date information. See section E.3.3 for details.

Exploitation: This field specifies exploitation information such as type of protection, use restriction and obligations to exploit an image. See section E.3.4 for details.

Identification: This field specifies an identifier of an image that is a link to a place where additional information is kept. See section E.3.5 for details.

Contact Point: This field specifies the contact point of the right holder. See section E.3.6 for details.

IPR History: This field specifies previous IPR metadata. See section E.3.7 for details.

E.3.1 Names

Names specify names related to the represented image IPR. These names include different categories, such as the creator, photographer, and producer, who may all claim rights. People appearing within the image may also be named, as there are restrictions on publishing the image of a person who has not consented to publication that varies from country to country. “Who,” “what” and “where” (i.e., the subject of the image) can also be names in the title of the image.

A name may be a person, an organization, or a reference to a name or a person. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="IPR_NAMES">
  <xsd:complexType>
    <xsd:choice                                                               maxOccurs="unbounded">
      <xsd:element ref="IPR_PERSON"/>
      <xsd:element ref="IPR_ORG"/>
      <xsd:element ref="IPR_NAME_REF"/>
    </xsd:choice>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="IPR_PERSON">
  <xsd:complexType base="dig35:tPerson" derivedBy="extension">
```

```
    <xsd:attribute name="DESCRIPTION"      type="xsd:string"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="IPR_ORG">
  <xsd:complexType base="dig35:tOrganization" derivedBy="extension">
    <xsd:attribute name="DESCRIPTION"      type="xsd:string"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="IPR_NAME_REF">
  <xsd:complexType base="xsd:string" derivedBy="extension">
    <xsd:attribute name="DESCRIPTION"      type="xsd:string"/>
  </xsd:complexType>
</xsd:element>
```

Person: This field specifies the person description. See F.2.13 for the format of this field.

Organization: This field specifies the organization description. See F.2.14 for the format of this field.

Name Reference: This field specifies a reference to a person or organization within the IPR metadata block.

Description: This field is the description of the name. The following values are suggested for this field and have the following meanings:

Table E-1: Suggested Name description values

Value	Meaning
`Original Work Author`	This value specifies that the field is the name of the author who created the original work that is represented in the image (e.g., painter, sculptor, architect, etc.), when the image is not a creation itself. By contrast, a photograph of a sunset will be considered as a creation of the photographer. Note that an original work author may be “anonymous.”
`Image Creator`	This value specifies that the field is the name of the image creator. The image creator could be, for example, the photographer who captured the original picture on film, the illustrator or graphic artist who conducted the image-creation process, etc.
`Right Holder`	This value specifies that the field is the name of the intellectual property right holder of the image. The right holder may be the author of the image, a stock photo agency, or vendor. He is the one to sell the licence to anyone willing to exploit the image, such as a publisher who will also sell the result or an end user in a pay-per-view process. The right holder has acquired specific rights from the creator or previous right holder in a transaction that usually has been registered officially.
`Represented Individuals`	This value specifies that the field is the name of an individual shown in the image. While it may be necessary to indicate the name of such individual as a description of the image, it is always useful to indicate the names of individuals appearing, because privacy rights may require that they grant consent to publish their image. In such an example, this descriptive field may result in restriction of use for the image, as well as describe the image contents.

```
Example:

<IPR_NAMES>
  <IPR_PERSON ID="01" DESCRIPTION="Image Creator">
    <PERSON_NAME>
      <NAME_COMP TYPE="Given">John</NAME_COMP>
      <NAME_COMP TYPE="Family">Doe</NAME_COMP>
    </PERSON_NAME>
  </IPR_PERSON>
  <IPR_ORG ID="0001" DESCRIPTION="Right Holder">
    <ORG_NAME>Example XYZ Inc.</ORG_NAME>
    <EMAIL>support@example-xyz.com</EMAIL>
  </IPR_ORG>

  <!--
     - Reference right holder ID="0001" defined above
    -->
  <IPR_NAME_REF DESCRIPTION="Represented Individual">0001</IPR_NAME_REF>
</IPR_NAMES>
```

E.3.2 Description

Description specifies the description of the content. It may be desirable to have a complementary explanation about the content of the image in order to exploit the content. For instance, a technical description of the content may help users in understanding and, therefore, valuing the content of an image (e.g., circumstances under which the image was taken). The format is vendor specific.

This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="IPR_DESCRIPTION">
  <xsd:complexType>
    <xsd:element name="IPR_TITLE"          type="dig35:tLangString"  minOccurs="0" maxOccurs="1"/>
    <xsd:element name="IPR_LEGEND"         type="dig35:tLangString"  minOccurs="0" maxOccurs="1"/>
    <xsd:element name="IPR_CAPTION"        type="dig35:tLangString"  minOccurs="0" maxOccurs="1"/>
    <xsd:element name="COPYRIGHT"          type="dig35:tLangString"  minOccurs="0" maxOccurs="1"/>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

Title: This field specifies the title of the image. It is a string that may be used, for instance, as a caption when printing. When the author creates the title, he may add meaning to the image. However, titles are not necessarily significant of IPR. This is determined on a case-by-case basis.

Legend: This field specifies the legend, a caption added to the picture, e.g., at the back of a photograph, written by the photographer to later classify the photos. It is generally a more detailed or technical description of what appears in the image. This field may answer the question, "why?" An example is saying, "image taken at dawn to test a 135 mm. zoom on stand."

Caption: This field specifies the caption of the image. This field addresses the text that has been added as complementary information to assist in understanding the image’s content (e.g., second draft by Durer for a study on a Biblical scene). The caption often has a tutorial motivation.

Copyright: This field specifies the copyright notice of the image. Usually this field defines the right holder who wants to be identified, saying e.g., "copyright agency XYZ." This is an indication that the property of the image is well defined and that the contact point is the designated agency.

Generally, the copyright notice is mandatory and must appear on the side of a photograph when printed or published.

```
Examples:

<IPR_DESCRIPTION>
  <IPR_TITLE>Small town in the far west</IPR_TITLE>
  <COPYRIGHT> &copy; Acme, Inc.</COPYRIGHT>
  <!-- "&copy;" represents © in XML -->
</IPR_DESCRIPTION>
```

E.3.3 Dates

Dates specify the IPR-related date information. There are a variety of valid DateTime Type formats. For example, a date may be an exact year, possibly with month and day, sometimes with hour, minute, second and thousandth (i.e., ISO timestamp, which is always GMT time). However, date may also be less delimiting. For example, the date could be “first half of the fifteenth century,” “late middle-age,” “early Roman,” etc.

Professional applications may prefer an exact date, whereas specifying a year +/- 5 years may satisfy users of early century photographs.

This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="IPR_DATES">
  <xsd:complexType>
    <xsd:element ref="IPR_DATE"                               maxOccurs="unbounded"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="IPR_DATE">
  <xsd:complexType base="dig35:tDateTime" derivedBy="extension">
    <xsd:attribute name="DESCRIPTION"     type="xsd:string"/>
    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

Date: The date field contains a date of arbitrary precision. See section F.2.8 for the format of this field. Note that the comment field defined in the DateTime Type may be used for describing more information on the field.

Description: This field is the description of the date. The precision of IPR dates may vary in accuracy depending on the age of the operation or item and other information known at the time of the metadata generation. The following values are suggested for this field and have the following meanings:

Table E-2: Suggested Date description values

Value	Meaning
`Original Work Creation`	This value specifies that the field is the date that the original work was created. All types of dates may appear here, as stated above.
`Picture Taken`	This value specifies that the field is the date that the picture was taken. Some digital cameras insert this information automatically.
`Scanned`	This value specifies that the field is the date that the image was scanned.
`Processed`	This value specifies that the field is the date that the image was processed.
`Modified`	This value specifies that the field is the date when any kind of modification was made to the original work. This field will store the most recent modification date. Although it is valid to have more than one modification date in this section, it would be more common that the entire IPR is updated during the modification, and the previous modifications moved to the IPR history. The processing tool may generate this date automatically.
`Last Modified`	This value specifies the last date the image was modified. This date should be easily found, because there may be either an automatic process filling this field and replacing the previous “last modification” as a “history field” or a manual process where the operator has to do the same operation by hand.

```
Examples:

<IPR_DATES>
  <IPR_DATE DESCRIPTION="Original Work Creation">
    <YEAR>1502</YEAR>
    <COMMENT>Original picture drawing date.</COMMENT>
  </IPR_DATE>
  <IPR_DATE DESCRIPTION="Original Work Creation">
    <DATE>2000-06-12</DATE>
    <COMMENT>Date the print was produced.</COMMENT>
  </IPR_DATE>
  <IPR_DATE DESCRIPTION="Scanned">
    <EXACT>2000-06-13T19:09:20</EXACT>
    <COMMENT>Drum scanner used.</COMMENT>
  </IPR_DATE>
  <IPR_DATE DESCRIPTION="Modified">
    <EXACT>2000-06-13T19:09:30</EXACT>
    <COMMENT>Wrinkles removed.</COMMENT>
  </IPR_DATE>
  <IPR_DATE DESCRIPTION="Modified">
    <EXACT>2000-06-13T19:09:31</EXACT>
    <COMMENT>Sofenned</COMMENT>
  </IPR_DATE>
</IPR_DATES>
```

E.3.4 Exploitation

Exploitation defines metadata to identify IPR protection mechanisms, specific restrictions imposed by the right holder or obligations resulting from the use of the image, and the IPR management system in use for this IPR metadata.

This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="IPR_EXPLOITATION">
  <xsd:complexType>
    <xsd:element name="IPR_PROTECTION"       type="dig35:tLangString"   minOccurs="0"   maxOccurs="1"/>
    <xsd:element name="IPR_USE_RESTRICTION"  type="dig35:tLangString"   minOccurs="0"   maxOccurs="1"/>
    <xsd:element name="IPR_OBLIGATION"       type="dig35:tLangString"   minOccurs="0"   maxOccurs="1"/>
    <xsd:element ref="IPR_MGMT_SYS"                                     minOccurs="0"   maxOccurs="1"/>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

Protection: This field specifies protection information. See section E.3.4.1 for details.

Use Restriction: This field specifies use restrictions. See section E.3.4.2 for details.

Obligation: This field specifies obligations of the image user. See section E.3.4.3 for details.

IPR Management System: This field specifies IPR management system information. See section E.3.4.4 for details.

E.3.4.1 Protection

Protection indicates that there is a watermark, that the image is registered, or that the image is protected by some other means. A value of zero specifies that the image is not protected and contains no watermark. Values between 1 and 255 are reserved for JPEG Utilities Registration Authority [JURA] use. Other values may exist (e.g., image registered by Acme Inc.). If this field is not present, then the watermark content (or its presence) is undefined.

```
Examples:

<!--
   - A protection scheme registered at JURA
  -->
<IPR_PROTECTION>1</IPR_PROTECTION>

<!--
   - The image is protected by a watermark system "PC03"
  -->
<IPR_PROTECTION>PC03</IPR_PROTECTION>

<!--
   - The high resolution image is encrypted
  -->
<IPR_PROTECTION>High resolution level encrypted</IPR_PROTECTION>

<!--
   - The image is not protected and contains no watermark
  -->
<IPR_PROTECTION>0</IPR_PROTECTION>
```

E.3.4.2 Use Restriction

Use restrictions may apply to an image that is not allowed outside the factory for industrial applications, or for which exclusive rights of copy have been delegated to a unique agency, or for which prior authorization of represented people is mandatory before publishing. Other restrictions may exist.

```
Example:

<IPR_USE_RESTRICTION>No copy</IPR_USE_RESTRICTION>
```

E.3.4.3 Obligation

Obligation may identify any mandatory condition for exploiting the content of a file. For example, the copyright information may be required to be written on the side of any printout for photographs; other obligations may concern the need to get allowance from persons represented on a picture if the picture is published. Obligations may vary with time. For example it may be forbidden to publish a photograph before a given date etc.

```
Example:

<IPR_OBLIGATION>Copyright mention mandatory</IPR_OBLIGATION>
```

E.3.4.4 IPR Management System

IPR Management Systems such as IPMP (Intellectual Property Management & Protection) or ECMS (Electronic Copyright Management System) use this fields to determine where information is kept regarding the management system. An example is to track the usage of an image.

During transfer, an agency determines the owner of the image from the management systems fields. It already knows the consumer, and uses this information to charge the user and credit the owner the amount as determined by the management system.

IPR Management System information is stored on a server describing the IPR of the image, and depending upon whether IPR licensing is mandatory or recommended, there must be a link to where such information is kept.

This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="IPR_MGMT_SYS">
  <xsd:complexType>
    <xsd:element name="IPR_MGMT_TYPE"         type="xsd:string"       minOccurs="0"    maxOccurs="1"/>
    <xsd:element name="IPR_MGMT_SYS_ID"       type="xsd:string"       minOccurs="0"    maxOccurs="1"/>
    <xsd:element name="IPR_MGMT_SYS_LOCATION" type="xsd:uriReference" minOccurs="0"    maxOccurs="1"/>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

IPR Management System Type: The IPR Management System being used.

IPR Management System ID: Information of an ID.

IPR Management System Location: Information of the location, e.g. URL.

E.3.5 Identification

Identification specifies a link to a place (e.g., secured database or other storage place) where critical information is kept. The identifier "identifies" specific content; therefore, if an image is cropped, modified or made into a new image, then the image must be registered again, and a new identifier must be acquired, because there are now two objects instead of merely one. The parent image must appear in the metadata set of the child.

This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="IPR_IDENTIFICATION">
  <xsd:complexType>
    <xsd:element ref="dig35:IPR_IDENTIFIER"                     minOccurs="0"maxOccurs="1"/>
    <xsd:element ref="dig35:LICENCE_PLATE"                      minOccurs="0"maxOccurs="1"/>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

Identifier: This field specifies the image identifier. See section E.3.5.1 for details.

Licence Plate: This field specifies licence plate. See section E.3.5.2 for details.

E.3.5.1 Generic IPR Identifier

Generic IPR Identifier specifies a generic purpose IPR identifier. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="IPR_IDENTIFIER">
  <xsd:complexType>
    <xsd:element name="IPR_ID_MODE"  type="dig35:tLangString" minOccurs="0"  maxOccurs="1"/>
    <xsd:element name="IPR_ID"       type="dig35:tLangString" minOccurs="0"  maxOccurs="1"/>
  </xsd:complexType>
</xsd:element>
```

Mode: This field specifies the identification mode.

ID: This field specifies the identification. The Mode field describes the content of this field.

```
Example:

<IPR_IDENTIFIER>
```

```
  <IPR_ID_MODE>ISBN</IPR_ID_MODE>
  <IPR_ID>0-330-28987</IPR_ID>
</IPR_IDENTIFIER>
```

E.3.5.2 Licence Plate

Licence Plate specifies the licence plate of the original image as defined in ISO 10918-3. The combination of the fields in the licence plate contains a globally unique identifying sequence of numbers.

This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="LICENCE_PLATE">
  <xsd:complexType>
    <xsd:element name="LP_COUNTRY"         type="xsd:string"      minOccurs="0"    maxOccurs="1"/>
    <xsd:element name="LP_REG_AUT"         type="xsd:string"      minOccurs="0"    maxOccurs="1"/>
    <xsd:element name="LP_REG_NUM"         type="xsd:string"      minOccurs="0"    maxOccurs="1"/>
    <xsd:element name="LP_DELIVERY_DATE"   type="xsd:timeInstant" minOccurs="0"    maxOccurs="1"/>
  </xsd:complexType>
</xsd:element>
```

Country: This field specifies the country of registration. The field contains the country code (3-digit number) for the licence plate as defined in ISO 3166-1 [10].

Registration Authority: This field specifies the registration authority number for the licence plate.

Registration Number: This field specifies the registration number for the licence plate.

Delivery Date: This field specifies when the Licence Plate was delivered to the registrant by the Registration Authority.

```
Example:

<LICENCE_PLATE>
  <LP_COUNTRY>FR</LP_COUNTRY>
  <LP_REG_AUT>1017</LP_REG_AUT>
  <LP_REG_NUM>1375</LP_REG_NUM>
</LICENCE_PLATE>
```

E.3.6 Contact Point

Contact Point specifies the contact point of the right holder. It includes a way to contact the current right holder in order to acquire the rights under the form of a licence. Such information may be a postal address, URL or any phone or fax number that is a non-ambiguous link to the right holder. Failing to fill this field would result in the impossibility to use the image, unless another field gives the link to the author.

A contact point may be a person, an organization, or a reference to a name or a person. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="IPR_CONTACT_POINT">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element ref="dig35:IPR_PERSON"/>
      <xsd:element ref="dig35:IPR_ORG"/>
      <xsd:element name="IPR_NAME_REF" type="xsd:string"/>
    </xsd:choice>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

Person: This field specifies the person description. See F.2.13 for the format of this field.

Organization: This field specifies the organization description. See F.2.14 for the format of this field.

Name Reference: This field specifies a reference to a person or organization within the IPR metadata block.

Description: This field is the description of the contact point. The following values are suggested for this field and have the following meanings:

Table E-3: Suggested Contact Point description values

Value	Meaning
`Collection`	This value is a link to a collector, museum, group, institution, etc.

Note that the contact point could potentially be a link to a name specified in the Names section of the IPR metadata.

```
Examples:

<!--
   - Example 1: Contact information embedded within the IPR Contact Point section
  -->
<IPR_CONTACT_POINT>
  <IPR_PERSON ID="10" DESCRIPTION="Image Creator">
     ...
  </IPR_PERSON>
</IPR_CONTACT_POINT>

<!--
   - Example 2: Contact information embedded within the IPR Contact Point section -->
  -->
<IPR_NAMES>
   <!-- Organization -->
   <IPR_ORG ID="10" DESCRIPTION="Right Holder">
      <!--
         -  Detail info of the organization that is the right holder of the content
        -->
   </IPR_ORG>
</IPR_NAMES>

<IPR_CONTACT_POINT>
  <!--  Reference a right holder specified in the IPR Name section -->
  <IPR_NAME_REF DESCRIPTION="Collection">10</IPR_NAME_REF>
</IPR_CONTACT_POINT>
```

E.3.7 IPR History

IPR History specifies previous IPR metadata. Each time an IPR information of an image is updated, some of the IPR metadata may be moved to the IPR History metadata section. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="IPR_HISTORY">
  <xsd:complexType>
    <xsd:element ref="dig35:IPR"                              minOccurs="0"  maxOccurs="unbounded"/>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>
```

IPR: This field specifies previous IPR information. See Annex E: for details.

E.4 Examples

Example:

```
<!--
   - Example:
   -    Original work author: John Doe, Sculptor
   -    Image creator: Jane Doe, Photographer
   -    Right holder: Example XYZ, Inc., support@example-xyz.com and www.example-xyz.com
  -->
<IPR>
  <IPR_NAMES>
    <IPR_PERSON ID="01" DESCRIPTION="Original Work Author">
      <PERSON_NAME>
        <NAME_COMP TYPE="Given">John</NAME_COMP>
        <NAME_COMP TYPE="Family">Doe</NAME_COMP>
      </PERSON_NAME>
      <COMMENT>Sculptor</COMMENT>
    </IPR_PERSON>
    <IPR_PERSON ID="10" DESCRIPTION="Image Creator">
      <PERSON_NAME>
        <NAME_COMP TYPE="Given">Jane</NAME_COMP>
        <NAME_COMP TYPE="Family">Doe</NAME_COMP>
      </PERSON_NAME>
      <COMMENT>Photographer</COMMENT>
    </IPR_PERSON>
    <IPR_ORG ID="0001" DESCRIPTION="Right Holder">
      <ORG_NAME>Example XYZ Inc.</ORG_NAME>
      <EMAIL>support@example-xyz.com</EMAIL>
      <WEB>www.example-xyz.com</WEB>
    </IPR_ORG>
  </IPR_NAMES>
  <IPR_DESCRIPTION>
    <IPR_TITLE>Small town in the far west</IPR_TITLE>
    <COPYRIGHT> &copy; Example XYZ, Inc.</COPYRIGHT>
    <!-- "&copy" represents © in XML -->
  </IPR_DESCRIPTION>
  <IPR_DATES>
    <IPR_DATE DESCRIPTION="Original Work Creation">
      <CENTURY>14</CENTURY>  <!-- Created in the 15th century (the 1400’s) -->
      <COMMENT>Early 15th century</COMMENT>
    </IPR_DATE>
    <IPR_DATE DESCRIPTION="Scanned">
      <EXACT>2000-06-13T19:09:20</EXACT>
      <COMMENT>Drum scanner used.</COMMENT>
    </IPR_DATE>
    <IPR_DATE DESCRIPTION="Modified">
      <EXACT>2000-06-13T19:09:31</EXACT>
      <COMMENT>Sofenned</COMMENT>
    </IPR_DATE>
  </IPR_DATES>
  <IPR_EXPLOITATION>
    <!--
       - A protection scheme registered at JURA
      -->
    <IPR_PROTECTION>1</IPR_PROTECTION>
    <IPR_USE_RESTRICTION>Low resolution only</IPR_USE_RESTRICTION>
    <IPR_OBLIGATION>Copyright mention mandatory</IPR_OBLIGATION>
  </IPR_EXPLOITATION>
  <IPR_IDENTIFICATION>
    <LICENCE_PLATE>
      <LP_COUNTRY>FR</LP_COUNTRY>
      <LP_REG_AUT>1017</LP_REG_AUT>
      <LP_REG_NUM>1375</LP_REG_NUM>
    </LICENCE_PLATE>
  </IPR_IDENTIFICATION>
  <IPR_CONTACT_POINT>
    <!--
       -  Reference a right holder specified in the IPR Name section XYZ, Inc. in this example
      -->
    <IPR_NAME_REF DESCRIPTION="Collection">0001</IPR_NAME_REF>
  </IPR_CONTACT_POINT>
</IPR>
```

Annex F: Fundamental Metadata Types and Fields

F.1 Overview

This Annex defines the Fundamental Types and Fields that are common to all blocks. The Types and Fields defined are intended only to be used or referred to in one of the other blocks, and is not a block of data in itself.

All fields listed in this document are optional unless otherwise stated.

The following diagram shows the relation of the Fundamental Metadata Types and Fields and various components of DIG35 metadata definition.

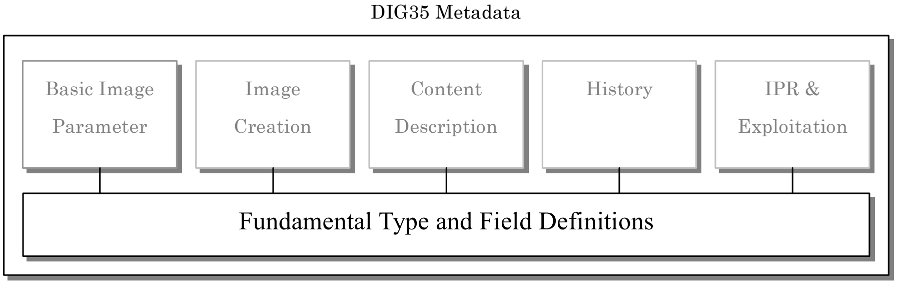


Figure F-1: High-level structure of the Fundamental Type and Field metadata

A DIG35 compliant metadata reader/editor shall understand all fields and their structures unless otherwise stated. It is recommended that a DIG35 metadata editor should persist information that is not understood when a DIG35 metadata file is modified.

F.2 DIG35 Defined Types and Fields

F.2.1 Non-negative Double Type

This type specifies a double number greater than or equal to zero.

```
Schema Definition:

<xsd:simpleType name="tNonNegativeDouble" base="xsd:double">
    <xsd:minInclusive value="0"/>
</xsd:simpleType>
```

F.2.2 Rational Type

This type specifies a rational number. It contains an enumerator and denominator in a single string.

```
Schema Definition:

<xsd:simpleType name="tRational" base="xsd:string">
    <xsd:pattern value="(\-|\+)?[0-9]+/[0-9]+"/>
</xsd:simpleType>
```

F.2.3 String Including Language Type

This type is used when a field requires a string and a language attribute definition. The content of this field is intended to store human readable data.

```
Schema Definition:

<xsd:complexType name="tLangString" base="xsd:string" derivedBy="extension">
  <xsd:attribute ref="xml:lang"/>
</xsd:complexType>
```

F.2.4 Degree Type

This type specifies a direction in degrees and fractions of degrees. The exact meaning of the values is dependent on usage.

```
Schema Definition:

<xsd:simpleType name="tDegree" base="xsd:double">
    <xsd:minExclusive value="-180"/>
    <xsd:maxInclusive value="180"/>
</xsd:simpleType>
```

F.2.5 Half Degree Type

This type specifies a direction in degrees and fractions of degrees. The exact meaning of the values is dependent on usage. This type defines a smaller range than Degree Type.

```
Schema Definition:

<xsd:simpleType name="tHalfDegree" base="xsd:double">
    <xsd:minExclusive value="-90"/>
    <xsd:maxInclusive value="90"/>
</xsd:simpleType>
```

F.2.6 Double Size Type

This field specifies a size in double coordinates.

```
Schema Definition:

<xsd:complexType name="tDoubleSize">
  <xsd:element name="WIDTH"          type="dig35:tNonNegativeDouble"/>
  <xsd:element name="HEIGHT"         type="dig35:tNonNegativeDouble"/>
</xsd:complexType>
```

F.2.7 Integer Size Type

This field specifies a size in integer coordinates (e.g. pixels).

```
Schema Definition:

<xsd:complexType name="tIntSize">
  <xsd:element name="WIDTH"          type="xsd:positiveInteger"/>
  <xsd:element name="HEIGHT"         type="xsd:positiveInteger"/>
</xsd:complexType>
```

F.2.8 DateTime Type

This field specifies a partial or exact date. A date can include either a specific day (e.g. 26 January 2000), or a more broad definition such as "Winter". A date may or may not include a time. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:complexType name="tDateTime">
  <xsd:choice                                                  minOccurs="0"   maxOccurs="1">
    <xsd:element name="EXACT"        type="xsd:timeInstant"/>
    <xsd:element name="DATE"         type="xsd:date"/>
    <xsd:sequence>
      <xsd:element name="MONTH"      type="tRecurringMonth"        minOccurs="0"   maxOccurs="1"/>
      <xsd:element name="YEAR"       type="xsd:year"               minOccurs="0"   maxOccurs="1"/>
      <xsd:element name="CENTURY"    type="xsd:century"            minOccurs="0"   maxOccurs="1"/>
    </xsd:sequence>
  </xsd:choice>

  <xsd:element name="WEEK_DAY"       type="dig35:tLangString"      minOccurs="0"   maxOccurs="1"/>
  <xsd:element name="SEASON"         type="dig35:tLangString"      minOccurs="0"   maxOccurs="1"/>

  <xsd:element ref="dig35:COMMENT"                                 minOccurs="0"   maxOccurs="1"/>

  <xsd:attribute ref="dig35:TIMESTAMP"/>
  <xsd:attribute ref="xml:lang"/>
</xsd:complexType>

<xsd:simpleType name="tRecurringMonth" base="xsd:positiveInteger">
  <xsd:minInclusive value="1"/>
  <xsd:maxInclusive value="12"/>
</xsd:simpleType>
```

Exact: This field contains an exact date and a time.

Date: This field contains a date (excluding the time of day).

Month: This field contains a month of the year. An integer value is used rather than a string to be consistent with the other fields contained in the DateTime Type. The value and month relation is defined in Table F-1.

Table F-1: Value-Month pair

Value	Month	Value	Month	Value	Month
1	January	5	May	9	September
2	February	6	June	10	October
3	March	7	July	11	November
4	April	8	August	12	December

Year: This field contains a calendar year. Positive values used for AD and negative values for BC. Note that the year zero is not valid.

Century: This field contains the century that an event occurred. For example, the twentieth century is stored as "19".

Week Day: This field is a text description of the day. Examples include: “Monday”, “Dienstag” etc.

Season: This field is a text description of a season. Examples include: “Spring”, “Summer”, “Autumn”, and “Winter.”

Comment: See section F.4.1 for details. Examples include “Easter Sunday”, “Morning”, “Just after lunch.”

```
Examples:

<EXACT>2000-08-15T13:20:00</EXACT>  <!-- An exact date and time -->

<DATE>2000-08-15</DATE>             <!-- A date (August 15, 2000) -->

<!--
   -  A month and year (August, 2000)
  -->
<MONTH>8</MONTH>
<YEAR>2000</YEAR>

<CENTURY>20</CENTURY>               <!-- The 21st century (the 2000's) -->

<!--
   -  A date and week day (Friday, August 15, 2000)
  -->
<DATE>2000-09-15</DATE>
<WEEK_DAY>Friday</WEEK_DAY>
<COMMENT>Opening of the Sydney 2000 Olympic games</COMMENT>
```

F.2.9 Address Type

The Address Type specifies the address of an object or location. For example, it may be used to describe the address an image was captured, or the address of the intellectual property owner of an image. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:complexType name="tAddress">
  <xsd:element name="ADDR_NAME"      type="dig35:tLangString"    minOccurs="0"  maxOccurs="1"/>
  <xsd:element ref="ADDR_COMP"                                   minOccurs="0"  maxOccurs="unbounded"/>
  <xsd:choice                                                    minOccurs="0"  maxOccurs="1">
    <xsd:element name="ZIPCODE"      type="xsd:string"/>
    <xsd:element name="POSTCODE"     type="xsd:string"/>
  </xsd:choice>
  <xsd:element name="COUNTRY"        type="dig35:tLangString"    minOccurs="0"  maxOccurs="1"/>

  <xsd:attribute name="TYPE"         type="xsd:string"/>
  <xsd:attribute ref="dig35:TIMESTAMP"/>
  <xsd:attribute ref="xml:lang"/>
</xsd:complexType>

<xsd:element name="ADDR_COMP">
  <xsd:complexType base="dig35:tLangString" derivedBy="extension">
    <xsd:attribute name="TYPE"       type="xsd:string"/>
  </xsd:complexType>
</xsd:element>
```

Address Name: It is a descriptive field for the address.

```
Example:

<ADDR_NAME>Yankee Stadium</ADDR_NAME>
```

Address Component: Multiple fields are used to specify the complete address. *The order of the address fields specifies the full address.* A full address shall be generated by concatenating the separate address fields**.**

For example, if the type is a `State`, this field contains the name of the state. Where the type is a `Street`, this field contains the name of the street. ISO 3166-2 lists country subdivision codes. These codes can optionally be used in this field, when the field is being used to specify a country subdivision.

Type: This is the name of this part of the address. Examples include "`Street`" or "`State`". ISO 3166-2 specifies country subdivisions and the types of these divisions. These subdivision types could optionally be used to specify the address type. Suggested values and their corresponding meanings are listed in Table F-2. Multiple values shall not be specified within a single field.

Table F-2: Address component type values

Value	Meaning
`Unit`	The unit number of the address to identify a house or a house name relative to a street.
`Room`	The room number within a building or an apartment.
`Street`	The street address in a postal address. Examples are street name, avenue and house number.
`Postbox`	The post office box number.
`City`	The locality of a geographic area.
`State`	The name of a geographical subdivision. Other terms such as `Province`, `Prefecture`, `County` may be used instead.

```
Examples:

<!--
   -  Address 1: 1, 23 Fourth Street, Carlingford, New South Wales
  -->
<ADDRESS>
  <ADDR_COMP TYPE="Unit">1</ADDR_COMP>
  <ADDR_COMP TYPE="Street">23 Fourth Street</ADDR_COMP>
  <ADDR_COMP TYPE="City">Carlingford</ADDR_COMP>
  <ADDR_COMP TYPE="State">New South Wales</ADDR_COMP>
</ADDRESS>

<!--
   -  Address 2: P.O.Box 333, New York, NY
  -->
<ADDRESS>
  <ADDR_COMP TYPE="Postbox">333</ADDR_COMP>
  <ADDR_COMP TYPE="City">New York</ADDR_COMP>
  <ADDR_COMP TYPE="State">NY</ADDR_COMP>
</ADDRESS>
```

Postcode / Zipcode: This field specifies the postcode (or zipcode) of the address. This field is not limited in length. The field has the title “Postcode” or “Zipcode”. Note an address shall not contain both a postcode and a zipcode.

Country: This field specifies the country of the address. The field can either contain the country code as defined in ISO 3166-1 [10] or a string identifying the country. The ISO 3166-1 country code is preferred.

Type: This field specifies the type of the whole address. The address type would include whether the address is a home address or a business address. Multiple type values may be specified delimited with a comma (“, ”).

Table F-3: Suggested Address type values

Value	Meaning
`Domestic`	The domestic delivery address.
`International`	The international delivery address.
`Postal`	The postal delivery address
`Home`	The delivery address for a residence.
`Work`	The delivery address for a place of work.

```
Example:

<!--
   - Home and work address
  -->
<ADDRESS TYPE="Home,Work">
  <ADDR_COMP TYPE="Street">1234 West Century Avenue</ADDR_COMP>
  <ADDR_COMP TYPE="City">Los Angles</ADDR_COMP>
  <ADDR_COMP TYPE="State">California</ADDR_COMP>
  <ZIPCODE>90044</ZIPCODE>
  <COUNTRY>US</COUNTRY>
</ADDRESS>
```

F.2.10 Phone Number Type

The Phone Number type specifies a phone number. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:complexType name="tPhone">
  <xsd:element name="COUNTRY_CODE"   type="xsd:string"        minOccurs="0"  maxOccurs="1"/>
  <xsd:element name="AREA"           type="xsd:string"        minOccurs="0"  maxOccurs="1"/>
  <xsd:element name="LOCAL"          type="xsd:string"        minOccurs="0"  maxOccurs="1"/>
  <xsd:element name="EXTENSION"      type="xsd:string"        minOccurs="0"  maxOccurs="1"/>

  <xsd:attribute name="TYPE"         type="xsd:string"/>
  <xsd:attribute ref="dig35:TIMESTAMP"/>
</xsd:complexType>
```

Country Code: This field contains the country code part of a phone number. This phone code does not include any prefix such as “00” used to dial international numbers, but instead just the international country code. This field also does not include a leading “+”.

Area Code: This field contains the local area code part of a phone number. This area code does not include leading zeros (or other digits) used to dial an interstate number from within a country. It appears as it would be appended directly to a country code.

Local Number: This field contains the local phone number.

Extension: This field contains the extension part of the phone number.

Type: This field defines the type of the phone number. The phone number type would include whether the phone number is a home phone number or a business phone number. Multiple type values may be specified delimited with a comma (“,”).

Table F-4: Suggested Phone type values

Value	Meaning
Home	Phone number associated with a residence.
Message	Phone number that has voice massage support.
Work	Phone number associated with a place of work.
Voice	Phone number to indicating a voice telephone.
Cell	Cellular telephone number.
Video	Videoconference telephone number.
BBS	Bulletin board system telephone number.
Modem	A modem connected telephone number.
Car	A car-phone telephone number.
ISDN	ISDN service telephone number.
PCS	Personal communication service telephone number.

```
Examples:

<!--
   - Example 1: Phone Number:(+61) 2 9212 2646
  -->
<PHONE>
  <COUNTRY_CODE>61</COUNTRY_CODE>
  <AREA>2</AREA>
  <LOCAL>92122646</LOCAL>
</PHONE>

<!--
   - Example 2:Work phone that has message support as well
  -->
<PHONE TYPE="Work,Message">
  <COUNTRY_CODE>61</COUNTRY_CODE>
  <AREA>2</AREA>
  <LOCAL>92122646</LOCAL>
</PHONE>
```

F.2.11 Email Address Type

The Email Type specifies an email address. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:complexType name="tEmail" base="dig35:tLangString" derivedBy="extension">
  <xsd:attribute name="TYPE"         type="xsd:string"/>
  <xsd:attribute ref="dig35:TIMESTAMP"/>
</xsd:complexType>
```

Type: This field contains the type of the email address.

```
Examples:

<EMAIL TYPE="Business">engineer@work.com</EMAIL>

<EMAIL TYPE="Home">cmb@home.com</EMAIL>
```

F.2.12 Web Address Type

The Web Type specifies a web address. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:complexType name="tWeb" base="dig35:tLangString" derivedBy="extension">
  <xsd:attribute name="TYPE"         type="xsd:string"/>
  <xsd:attribute ref="dig35:TIMESTAMP"/>
</xsd:complexType>
```

Type: This field contains the type of the web page.

```
Examples:

<WEB TYPE="Business">http://www.example.com/</WEB>

<WEB TYPE="Home">http://www.home.com/~personal</WEB>
```

F.2.13 Person Type

The Person Type specifies a person. The sub-fields are compatible with the vCard description defined in RFC 2426 [9]. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:complexType name="tPerson">
  <xsd:element name="NAME_TITLE"     type="dig35:tLangString" minOccurs="0"  maxOccurs="1"/>
  <xsd:element ref="PERSON_NAME"                              minOccurs="0"  maxOccurs="unbounded"/>
  <xsd:element name="NICKNAME"       type="dig35:tLangString" minOccurs="0"  maxOccurs="unbounded"/>
  <xsd:element name="JOB_TITLE"      type="dig35:tLangString" minOccurs="0"  maxOccurs="1"/>
  <xsd:choice                                                 minOccurs="0"  maxOccurs="1">
    <xsd:element name="PERSON_ORG"   type="dig35:tOrganization"/>
    <xsd:element name="ORG_REF"      type="xsd:string"/>
  </xsd:choice>
  <xsd:element name="ADDRESS"        type="dig35:tAddress"    minOccurs="0"  maxOccurs="unbounded"/>
  <xsd:element name="PHONE"          type="dig35:tPhone"      minOccurs="0"  maxOccurs="unbounded"/>
  <xsd:element name="EMAIL"          type="dig35:tEmail"      minOccurs="0"  maxOccurs="unbounded"/>
  <xsd:element name="WEB"            type="dig35:tWeb"        minOccurs="0"  maxOccurs="unbounded"/>
  <xsd:element name="BIRTH_DATE"     type="xsd:date"          minOccurs="0"  maxOccurs="1"/>
  <xsd:element name="AGE"            type="xsd:timeDuration"  minOccurs="0"  maxOccurs="1"/>
  <xsd:element ref="dig35:COMMENT"                            minOccurs="0"  maxOccurs="1"/>

  <xsd:attribute name="ID"           type="xsd:string"/>
  <xsd:attribute ref="dig35:TIMESTAMP"/>
  <xsd:attribute ref="xml:lang"/>
</xsd:complexType>

<xsd:element name="PERSON_NAME">
  <xsd:complexType>
    <xsd:element ref="NAME_COMP"                                             maxOccurs="unbounded"/>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="NAME_COMP">
  <xsd:complexType base="xsd:string" derivedBy="extension">
    <xsd:attribute name="TYPE">
      <xsd:simpleType base="xsd:string">
        <xsd:enumeration value="Prefix"/>
        <xsd:enumeration value="Given" use="default" value="Given"/>
        <xsd:enumeration value="Family"/>
        <xsd:enumeration value="Suffix"/>
        <xsd:enumeration value="Maiden"/>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
```

ID: This field specifies the unique identifier for the person.

Name Title: The field contains the person's title.

```
Examples:

<NAME_TITLE>Duke of York</NAME_TITLE>

<NAME_TITLE>President of Australia</NAME_TITLE>
```

Person Name: This field defines a framework to describe a person's name. A person's name is composed of multiple name components (e.g. given name(s) and family name(s)). *The order of the name component fields specifies the full name of the person.* For example, in languages where the family name is usually placed before the given name, then they would appear in this order in the file.

Name Component: This field contains a single portion (word) of the name of a person. A name component field can contain a single initial rather than a complete word. To specify the full name of a person, multiple name component fields are used. This field contains a type as specified below.

Type: This field defines the type of the Name Component. This field would include whether the name component is a Suffix, Prefix, Given or Family name. Multiple values shall not be specified within a single type filed.

Table F-5: Suggested Name Component type values

Value	Meaning
`Prefix`	A personal title. (e.g. Dr., Sir)
`Given`	A name construct that is normally given to an individual by the parent or is chosen by the individual. This is the default value of the name component type.
`Family`	A name component that is normally inherited by their parent or assumed by marriage.
`Suffix`	A generation qualifier (e.g. Jr., III), decorations and awards. (e.g. Q.C., Ph. D)
`Maiden`	A name component of a woman's family name before getting married.

```
Examples:

<!--
   - Example name 1: Rev Dr Chuck Berry III Esq
  -->
<PERSON_NAME>
  <NAME_COMP TYPE="Prefix">Rev</NAME_COMP>
  <NAME_COMP TYPE="Prefix">Dr</NAME_COMP>
  <NAME_COMP TYPE="Given">Chuck</NAME_COMP>
  <NAME_COMP TYPE="Family">Berry</NAME_COMP>
  <NAME_COMP TYPE="Suffix">III</NAME_COMP>
  <NAME_COMP TYPE="Suffix">Esq</NAME_COMP>
</PERSON_NAME>

<!--
   - Example name 2: 石井 克己
  -->
<PERSON_NAME xml:lang="ja">
  <NAME_COMP TYPE="Family">石井</NAME_COMP>
  <NAME_COMP TYPE="Given">克己</NAME_COMP>
</PERSON_NAME>
<PERSON_NAME xml:lang="en">
  <NAME_COMP TYPE="Given">Katsuki</NAME_COMP>
  <NAME_COMP TYPE="Family">Ishii</NAME_COMP>
</PERSON_NAME>
```

Nickname: This field specifies a nickname of the person, e.g. "Jimmy."

Job Title: This field specifies the person's job title.

```
Examples:

<JOB_TITLE>Engineer</JOB_TITLE>

<JOB_TITLE>Brain Surgeon</JOB_TITLE>

<JOB_TITLE>Racing Car Driver</JOB_TITLE>
```

Organization: This field specifies the organization for which a person is a member of. The organization field may be either contained within the person field, or referenced:

Organization: Organization description. See section F.2.14 for the format of this field.

Organization Reference: A reference to the organization. This field is a link to one of the Organization fields within the metadata.

Address: This field specifies address information for the person. For example, it can contain a home address or a work address. It does not necessarily contain the address depicted within the image, but instead information about the person. See section F.2.9 for the format of this field.

Phone Number: This field specifies phone number information for the person. See section F.2.10 for the format of this field.

Email Address: This field specifies an email address for a person. See section F.2.11 for the format of this field.

Web Page: This field specifies a web page for a person. See section F.2.12 for the format of this field.

Date of Birth: This field specifies the birth date of the person. This field specifies an exact date. For non-specific information the comment field is used.

Age: This field specifies the age of a person.

```
Examples:

<AGE>P10Y</AGE>       <!-- 10 years -->
<AGE>P2Y3M</AGE>      <!-- 2 years, 3 months -->
<AGE>P6M</AGE>        <!-- 6 months -->
<AGE>P16D</AGE>       <!-- 16 days -->
```

Comment: This field specifies user- and/or application-defined information beyond the scope of other properties in the person field. See section F.4.1 for the format of this field.

F.2.14 Organization Type

The Organization Type specifies an organization. The sub-fields are compatible with the vCard description defined in RFC 2426 [9]. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:complexType name="tOrganization">
  <xsd:element name="ORG_NAME"       type="dig35:tLangString" minOccurs="0"  maxOccurs="1"/>
  <xsd:element name="ADDRESS"        type="dig35:tAddress"    minOccurs="0"  maxOccurs="unbounded"/>
  <xsd:element name="PHONE"          type="dig35:tPhone"      minOccurs="0"  maxOccurs="unbounded"/>
  <xsd:element name="EMAIL"          type="dig35:tEmail"      minOccurs="0"  maxOccurs="unbounded"/>
  <xsd:element name="WEB"            type="dig35:tWeb"        minOccurs="0"  maxOccurs="unbounded"/>
  <xsd:element ref="dig35:COMMENT"                            minOccurs="0"  maxOccurs="1"/>

  <xsd:attribute name="ID"           type="xsd:string"/>
  <xsd:attribute ref="dig35:TIMESTAMP"/>
  <xsd:attribute ref="xml:lang"/>
</xsd:complexType>
```

ID: This field specifies the unique identifier for the organization.

Organization Name: This field specifies the name of the organization.

Address: This field specifies address information for the organization. It does not necessarily contain the address depicted within the image, but instead information about the organization. See section F.2.9 for the format of this field.

Phone Number: This field specifies phone number information for the organization. See section F.2.10 for the format of this field.

Email Address: This field specifies an email address for an organization. See section F.2.11 for the format of this field.

Web Page: This field specifies a web page for an organization. See section F.2.12 for the format of this field.

Comment: This field specifies user- and/or application-defined information beyond the scope of other properties in the organization field. See section F.4.1 for the format of this field.

F.2.15 Location Type

The Location Type specifies the physical location of an object or a scene. For example, it may be used to describe an object within an image, or the location of a camera at the time of capture.

The Location is the physical location, whereas the Position is the position of an object relative to the image.

Examples of locations:
Under the table
Longitude: 10 degrees, Latitude: 20 degrees

Example of positions:
Top left corner
Rectangle: x=0.1, y=0.1, width=0.2, height=0.3

This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:complexType name="tLocation">
  <xsd:element ref="dig35:COORD_LOC"                            minOccurs="0"   maxOccurs="1"/>
  <xsd:element name="ADDRESS"        type="dig35:tAddress"     minOccurs="0"   maxOccurs="1"/>
  <xsd:element ref="dig35:GPS"                                  minOccurs="0"   maxOccurs="1"/>
  <xsd:element ref="dig35:COMMENT"                              minOccurs="0"   maxOccurs="1"/>

  <xsd:attribute ref="dig35:TIMESTAMP"/>
  <xsd:attribute ref="xml:lang"/>
</xsd:complexType>
```

Coordinate Location: This field specifies the exact longitude, latitude and altitude of an object. See section F.2.15.1 for details.

Address: This field specifies the location of an object using an address. See section F.2.9 for the format of this field.

GPS: This field specifies location information received from a GPS receiver. See section F.2.15.2 for details.

Comment: This field specifies the location of an object that cannot be described using the other location fields. See section F.4.1 for the format of this field. Example: "Under the table"

F.2.15.1 Coordinate Location

The Coordinate Location specifies the terrestrial location (altitude / longitude / latitude) of an object. It may be used to describe the content of an image along with the location of a camera.

While the Coordinate Location may have come from a GPS (and a GPS block may or may not be present in the metadata), the values in the coordinate location may have come for some other means. For this reason, the location information is a more general system for storing the location than the GPS system. The location information and the raw GPS data are stored in different formats.

GPS is one of a number of methods that may be used to determine a location. If the GPS information is filled in, it is expected that the coordinate location is also specified. A reader must only look in a single place to determine the coordinate location (this field).

The meridian through Greenwich (Great Britain) is defined with the value longitude l =0. The longitude l of a point P on the surface is the angle between the planes through its meridian and the Greenwich meridian. The longitude is counted from Greenwich up to l =±180° in east(+) and west(-) directions.

The latitude j of a point P is the angle between a line normal to its parallel and the equatorial plane (j =0). On a sphere this normal line will be the connecting line between its center and the point P. On the elliptical earth this line will only pass the center if P is situated at the equator. The latitude is counted from the equator up to j =±90° in north (+) and south (-) directions.

This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="COORD_LOC">
  <xsd:complexType>
    <xsd:element name="LONGITUDE"    type="dig35:tDegree"     minOccurs="0"   maxOccurs="1"/>
    <xsd:element name="LATITUDE"     type="dig35:tHalfDegree" minOccurs="0"   maxOccurs="1"/>
    <xsd:element name="ALTITUDE"     type="xsd:double"        minOccurs="0"   maxOccurs="1"/>

  <xsd:attribute ref="dig35:TIMESTAMP"/>
  </xsd:complexType>
</xsd:element>
```

Longitude: This field contains the longitude, represented in double degrees and fractions of degrees, e.g. “138.700”, “-122.450”.

Latitude: This field contains the latitude, represented in double degrees and fractions of degrees, e.g. “35.383”, “37.767”.

Altitude: This field would contain the distance in meters. Zero is sea level, positive is above, and negative is below.

F.2.15.2 Raw GPS Information

The information in these fields is expected to be imported from a GPS system and is compatible with NMEA-0138 [24]. For this reason, the fields are not consistent with other DIG35 metadata fields. For example, a distance on the GPS fields may be stored in miles, while all other metadata distances are stored in meters. These fields are compatible with Exif version 2.1.

If information for latitude, longitude and altitude are present in the raw GPS information, the matching fields in the Coordinate Location must be filled in.

This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="GPS">
  <xsd:complexType>

    <xsd:element name="GPS_LAT_REF"                                         minOccurs="0"  maxOccurs="1">
      <xsd:simpleType base="xsd:string">
        <xsd:enumeration value="N"/>
        <xsd:enumeration value="S"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="GPS_LATITUDE"      type="tDms"                       minOccurs="0"  maxOccurs="1"/>

    <xsd:element name="GPS_LONG_REF"                                        minOccurs="0"  maxOccurs="1">
      <xsd:simpleType base="xsd:string">
        <xsd:enumeration value="E"/>
        <xsd:enumeration value="W"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="GPS_LONGITUDE"     type="tDms"                       minOccurs="0"  maxOccurs="1"/>

    <xsd:element name="GPS_ALTITUDE"      type="dig35:tNonNegativeDouble"  minOccurs="0"  maxOccurs="1"/>

    <xsd:element name="GPS_TIME"          type="xsd:timeInstant"            minOccurs="0"  maxOccurs="1"/>

    <xsd:element name="GPS_SATELLITES"    type="xsd:string"                 minOccurs="0"  maxOccurs="1"/>
    <xsd:element name="GPS_STATUS"                                          minOccurs="0"  maxOccurs="1">
      <xsd:simpleType base="xsd:string">
        <xsd:enumeration value="A"/>
        <xsd:enumeration value="V"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="GPS_MEASURE_MODE"                                    minOccurs="0"  maxOccurs="1">
```

```
      <xsd:simpleType base="xsd:positiveInteger">
        <xsd:minExclusive value="2"/>
        <xsd:maxInclusive value="3"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="GPS_DOP"           type="dig35:tNonNegativeDouble"  minOccurs="0"  maxOccurs="1"/>

    <xsd:element name="GPS_SPEED_REF"                                      minOccurs="0"  maxOccurs="1">
      <xsd:simpleType base="xsd:string">
        <xsd:enumeration value="K"/>
        <xsd:enumeration value="N"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="GPS_SPEED"         type="dig35:tNonNegativeDouble"  minOccurs="0"  maxOccurs="1"/>

    <xsd:element name="GPS_TRACK_REF"                                      minOccurs="0"  maxOccurs="1">
      <xsd:simpleType base="xsd:string">
        <xsd:enumeration value="T"/>
        <xsd:enumeration value="M"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="GPS_TRACK"         type="dig35:tNonNegativeDouble"  minOccurs="0"  maxOccurs="1"/>

    <xsd:element name="GPS_IMAGE_DIR_REF"                                  minOccurs="0"  maxOccurs="1">
      <xsd:simpleType base="xsd:string">
        <xsd:enumeration value="T"/>
        <xsd:enumeration value="M"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="GPS_IMAGE_DIR"     type="dig35:tNonNegativeDouble"  minOccurs="0"  maxOccurs="1"/>

    <xsd:element name="GPS_MAP_DATUM"     type="xsd:string"                minOccurs="0"  maxOccurs="1"/>

    <xsd:element name="GPS_DEST_LAT_REF"                                   minOccurs="0"  maxOccurs="1">
      <xsd:simpleType base="xsd:string">
        <xsd:enumeration value="N"/>
        <xsd:enumeration value="S"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="GPS_DEST_LATITUDE" type="tDms"                      minOccurs="0"  maxOccurs="1"/>

    <xsd:element name="GPS_DEST_LONG_REF"                                  minOccurs="0"  maxOccurs="1">
      <xsd:simpleType base="xsd:string">
        <xsd:enumeration value="E"/>
        <xsd:enumeration value="W"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="GPS_DEST_LONGITUDE" type="tDms"                     minOccurs="0"  maxOccurs="1"/>

    <xsd:element name="GPS_DEST_BEARING_REF"                               minOccurs="0"  maxOccurs="1">
      <xsd:simpleType base="xsd:string">
        <xsd:enumeration value="T"/>
        <xsd:enumeration value="M"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="GPS_DEST_BEARING"  type="dig35:tNonNegativeDouble"  minOccurs="0"  maxOccurs="1"/>

    <xsd:element name="GPS_DEST_DISTANCE_REF"                              minOccurs="0"  maxOccurs="1">
      <xsd:simpleType base="xsd:string">
        <xsd:enumeration value="K"/>
        <xsd:enumeration value="N"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="GPS_DEST_DISTANCE" type="dig35:tNonNegativeDouble"  minOccurs="0"  maxOccurs="1"/>
```

```
  </xsd:complexType>
</xsd:element>

<xsd:complexType name="tDms">
  <xsd:element name="D"                    type="xsd:nonNegativeInteger"/>
  <xsd:element name="M"                    type="xsd:nonNegativeInteger"/>
  <xsd:element name="S"                    type="dig35:tNonNegativeDouble" minOccurs="0" maxOccurs="1"/>
</xsd:complexType>
```

GPS Latitude Reference: This field specifies whether the GPS Latitude is North or South.

Table F-6: GPS Latitude Reference values

Value	Meaning
N	North Latitude
S	South Latitude

GPS Latitude: This field contains the latitude of the GPS receiver.

D: The number of degrees of latitude.
M: The number of minutes of latitude.
S: The number of seconds of latitude.

GPS Longitude Reference: This field specifies whether the GPS Longitude is East or West.

Table F-7: GPS Longitude Reference values

Value	Meaning
E	East Longitude
W	West Longitude

GPS Longitude: This field contains the longitude of the GPS receiver.

D: The number of degrees of longitude.

M: The number of minutes of longitude.

S: The number of seconds of longitude.

GPS Altitude: This field contains the altitude of the GPS receiver. The altitude reading is given in meters relative to sea level (geoid).

GPS Time: This field contains the time of the GPS location was determined. This field is in Greenwich Mean Time. Note this is not necessarily the camera capture time.

GPS Satellites: This field contains information about the satellites used to determine the camera position. This tag can be used to describe the number of satellites, their ID number, angle of elevation, azimuth, SNR and other information. The format is not specified.

GPS Status: This field contains information on the GPS receiver at time of image capture.

Table F-8**:** GPS Status values

Value	Meaning
A	Measurement is in progress
V	Measurement is interrupted

GPS Measure Mode: This field contains information on the measurement mode used to determine the GPS location.

Table F-9: GPS Measure Mode values

Value	Meaning
2	2 dimensional measurement
3	3 dimensional measurement

GPS DOP: This field contains a value indicating the GPS DOP (data degree of precision). An HDOP (horizontal degree of precision) value is written during a two-dimensional measurement, and a PDOP (3D degree of precision) value is written during a three-dimensional measurement.

GPS Speed Reference: This field contains the units of measure for the GPS Speed field.

Table F-10: GPS Speed Reference unit values

Value	Meaning
K	Kilometers per hour
N	Knots

GPS Speed: This field contains a value indicating the speed of the GPS receiver. The value units are defined by the GPS Speed Reference.

GPS Track Reference: This field contains the reference for the GPS Track field.

Table F-11: GPS Track Reference values

Value	Meaning
T	True North
M	Magnetic North

GPS Track: This field contains the value in degrees indicating the direction of the GPS receiver movement. 0 indicates North and 90 indicate East.

GPS Image Direction Reference: This field contains the reference for the GPS Image Direction field.

Table F-12: GPS Image Direction Reference values

Value	Meaning
T	True North
M	Magnetic North

GPS Image Direction: This field contains the value in degrees indicating the direction the camera is facing at the time of taking the picture. 0 indicates North and 90 indicate East.

GPS Map Datum: This field specifies the geodetic survey data used by the GPS receiver. For example, if the survey data is restricted to Japan, the value of this tag is “TOKYO” or “WSG-84”.

GPS Destination Latitude Reference: This field specifies whether the GPS Destination Latitude is North or South.

Table F-13: GPS Destination Latitude Reference values

Value	Meaning
N	North Latitude
S	South Latitude

GPS Destination Latitude: This field contains the destination latitude of the GPS receiver.

D: The number of degrees of latitude.

M: The number of minutes of latitude.

S: The number of seconds of latitude.

GPS Destination Longitude Reference: This field specifies whether the GPS Destination Longitude is East or West.

Table F-14: GPS Destination Longitude Reference values

Value	Meaning
E	East Longitude
W	West Longitude

GPS Destination Longitude: This field contains the destination longitude of the GPS receiver.

D: The number of degrees of longitude.

M: The number of minutes of longitude.

S: The number of seconds of longitude.

GPS Destination Bearing Reference: This field contains the reference for the GPS Destination Bearing field.

Table F-15: GPS Destination Bearing Reference values

Value	Meaning
T	True North
M	Magnetic North

GPS Destination Bearing: This field contains the value in degrees indicating the direction of the destination from the GPS receiver. 0 indicates North and 90 indicate East.

GPS Destination Distance Reference: This field contains the units of measure for the GPS Destination Distance field.

Table F-16: GPS Destination Distance units

Value	Meaning
K	Kilometers
N	Nautical Miles

GPS Destination Distance: This field contains a value indicating the distance to the destination from the GPS receiver. The value units are defined by the GPS Destination Distance Reference.

F.2.16 Direction Type

The Direction type specifies a three-dimensional heading. While this type is primarily used to specify the direction a camera is facing, it could also be used to specify information about an object in a scientific photograph for example. When calculating the direction the camera is facing, first the yaw is applied, then the pitch, then the roll.

The fields in this section have been taken from [23]. This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:complexType name="tDirection">
  <xsd:element name="YAW"            type="dig35:tDegree"     minOccurs="0"  maxOccurs="1"/>
  <xsd:element name="PITCH"          type="dig35:tHalfDegree" minOccurs="0"  maxOccurs="1"/>
  <xsd:element name="ROLL"           type="dig35:tDegree"     minOccurs="0"  maxOccurs="1"/>
  <xsd:element ref="dig35:COMMENT"                            minOccurs="0"  maxOccurs="1"/>

  <xsd:attribute ref="dig35:TIMESTAMP"/>
  <xsd:attribute ref="xml:lang"/>
</xsd:complexType>
```

Yaw: This field is the direction the capture device is facing. The field is measured in degrees. North is 0, East is 90, South is 180 and West is -90.

Pitch: This field is a measure of the elevation angle of the capture device. This field is a `Double` value between –90 and +90, also measured in degrees. 0 facing horizontal. 90 is facing vertically straight upwards, and –90 vertically downwards.

Roll: This field is a measure of the rotation angle of the capture device. This field is a `Double` value between –180 and 180, also measured in degrees. 0 facing horizontal. 90 where the device is rotated clockwise and the left of the device is facing upwards, and –90 where the device is rotated anti-clockwise. 180 is upside down.

Comment: This field specifies user- and/or application-defined information beyond the scope of other properties in the direction field. See section F.4.1 for the format of this field. Example: "Upwards", "To the left"

F.2.17 Position Type

The Position Type specifies the position of an object, within an image. The Position Type can be one of the following:

- An `x`, `y` single point.
- A rectangular area (specified as an `x`, `y`, `width` and `height`)
- A set of splines that represent an area of the image.
- A free-text comment field

The image is described in a Cartesian system, with the X-axis horizontal and pointing to the right, the Y-axis vertical and pointing downward, and the origin at the upper left corner. The scale is such that the height of the image is normalized to 1.0. To keep the scale of the X-axis and the Y-axis the same, the image width (R) is its aspect ratio (width/height). Thus, a square part of any image has equal width and height in this coordinate system. (See Figure F-2)

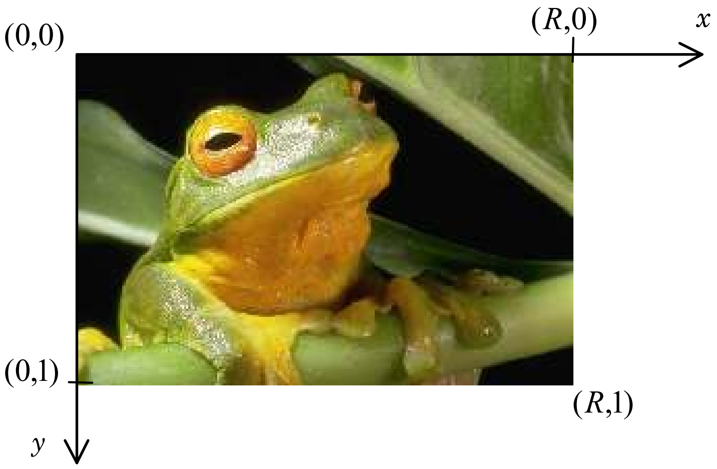


Figure F-2: Coordinate system

Note that this information may become useless if the image is cropped or manipulated. See Location for the difference between the Position and Location types.

This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:complexType name="tPosition">
  <xsd:choice                                             minOccurs="0"  maxOccurs="1">
    <xsd:element name="POINT"        type="dig35:tPoint"/>
    <xsd:element name="RECT"         type="dig35:tRect"/>
    <xsd:sequence>
      <xsd:element name="RECT"       type="dig35:tRect"/>
      <xsd:element name="REGION"     type="dig35:tRegion"/>
    </xsd:sequence>
  </xsd:choice>
  <xsd:element ref="dig35:COMMENT"                        minOccurs="0"  maxOccurs="1"/>

  <xsd:attribute ref="dig35:TIMESTAMP"/>
</xsd:complexType>
```

Single Point: This field specifies a single point in the coordinate system defined in Figure F-2. See section F.2.17.1 for details.

Rectangular Region: This field specifies a rectangular region in the coordinate system defined in Figure F-2. See section F.2.17.2 for details.

Arbitrary Region: This field specifies arbitrary region. It contains a start point and one or more splines in the coordinate system defined in Figure F-2. See section F.2.17.3 for details.

Comment: This field specifies the position of an object less accurately than one of the above methods. For example, this field may contain "Bottom left-hand corner" or "Second from the left in the top row". See section F.4.1 for the format of this field.

F.2.17.1 Point Type

This section specifies details about a single point on an image. This field is used to describe a single point in the coordinate system defined in Figure F-2. This field shall contain the sub-fields listed below.

```
Schema Definition:

<xsd:complexType name="tPoint">
  <xsd:element name="X"              type="dig35:tNonNegativeDouble"/>
  <xsd:element name="Y"              type="dig35:tNonNegativeDouble"/>
</xsd:complexType>
```

X: This field specifies the X coordinate of the point.

Y: This field specifies the Y coordinate of the point.

```
Example:

<POINT>
  <X>0.2</X>
  <Y>0.5</Y>
</POINT>
```

F.2.17.2 Rectangle Type

This section specifies details about a rectangular region on an image. This field is used to describe a rectangular region in the coordinate system defined in Figure F-2. See section F.2.17.1 for the format of this field. Additionally, this field shall contain the sub-fields listed below.

```
Schema Definition:

<xsd:complexType name="tRect" base="dig35:tPoint" derivedBy="extension">
  <xsd:element name="WIDTH"          type="dig35:tNonNegativeDouble"/>
  <xsd:element name="HEIGHT"         type="dig35:tNonNegativeDouble"/>
</xsd:complexType>
```

X: The left of the rectangle.

Y: The top of the rectangle.

Width: The width of the rectangle (to the right of x).

Height: The height of the rectangle (below y).

```
Example:

<RECT>
  <X>0.2</X>
  <Y>0.2</Y>
  <WIDTH>0.1</WIDTH>
  <HEIGHT>0.4</HEIGHT>
</RECT>
```

F.2.17.3 Region Type

This section specifies details about an arbitrary region on an image. This field consists of a start point and one or more segments. Each segment may be either a straight line (specified using a point), or a spline.

Where an arbitrary region is specified, a Rectangular Region must also be specified (which is the bounding box of the Arbitrary Region). A standard DIG35-compliant metadata reader or editor has the option of not using the Arbitrary Region, even if the Rectangular Region is used.

This field shall contain the sub-fields listed below.

```
Schema Definition:

<xsd:complexType name="tRegion">
  <xsd:element name="POINT"          type="dig35:tPoint"/>
  <xsd:choice                                                  minOccurs="0"  maxOccurs="unbounded">
    <xsd:element name="POINT"        type="dig35:tPoint"/>
    <xsd:element ref="SPLINE"/>
  </xsd:choice>
</xsd:complexType>

<xsd:element name="SPLINE">
  <xsd:complexType>
    <xsd:element name="X1"  type="dig35:tNonNegativeDouble"/>
    <xsd:element name="Y1"  type="dig35:tNonNegativeDouble"/>
    <xsd:element name="X2"  type="dig35:tNonNegativeDouble"/>
    <xsd:element name="Y2"  type="dig35:tNonNegativeDouble"/>
    <xsd:element name="X"   type="dig35:tNonNegativeDouble"/>
    <xsd:element name="Y"   type="dig35:tNonNegativeDouble"/>
  </xsd:complexType>
</xsd:element>
```

Start Point: This is the starting point of the spline in the coordinate system defined in Figure F-2. See F.2.17.1 for the format of this field.

Point: This field specifies a line starting at the end of the previous spline and ending at the new point. See F.2.17.1 for the format of this field.

Spline: This field specifies defines a bezier curve starting at the end of the previous spline, and ending at the new end point (x, y), with x1, y1 and x2, y2 being the first and second control points of the spline respectively.

```
Example:

<!--
   - A circle spline example
  -->
<REGION>
  <!--
     - Starting point. Far left of the circle.
    -->
  <POINT>
    <X>0</X>
    <Y>0.5</Y>
  </POINT>

  <!--
     - Curve to the top of the circle.
    -->
  <SPLINE>
    <X1>0</X1>
    <Y1>0.2239</Y1>
    <X2>0.2239</X2>
    <Y2>0</Y2>
    <X>0.5</X>
    <Y>0</Y>
  </SPLINE>

  <!--
     - Curve to the far right of the circle.
    -->
  <SPLINE>
    <X1>0.7761</X1>
    <Y1>0</Y1>
    <X2>1</X2>
    <Y2>0.2239</Y2>
    <X>1</X>
    <Y>0.5</Y>
  </SPLINE>

  <!--
     - Curve to the bottom of the circle.
    -->
  <SPLINE>
    <X1>1</X1>
    <Y1>0.7761</Y1>
    <X2>0.7761</X2>
    <Y2>1</Y2>
    <X>0.5</X>
    <Y>1</Y>
  </SPLINE>

  <!--
     - Curve to the far left of the circle.
    -->
  <SPLINE>
    <X1>0.2239</X>
    <Y1>1</Y1>
    <X2>0</X2>
    <Y2>0.7761</Y2>
    <X>0</X>
    <Y>0.5</Y>
  </SPLINE>

</REGION>
```

F.2.18 Product Details Type

This section specifies details about a product (hardware or software). By combining the sub-fields of the product details, a unique value shall be created.

This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:complexType name="tProductDetails">
  <xsd:element name="MANUFACTURER"   type="dig35:tOrganization"  minOccurs="0"    maxOccurs="1"/>
  <xsd:element name="MODEL"          type="xsd:string"           minOccurs="0"    maxOccurs="1"/>
  <xsd:element name="SERIAL"         type="xsd:string"           minOccurs="0"    maxOccurs="1"/>
  <xsd:element name="VERSION"        type="xsd:string"           minOccurs="0"    maxOccurs="1"/>

  <xsd:attribute ref="dig35:TIMESTAMP"/>
  <xsd:attribute ref="xml:lang"/>
</xsd:complexType>
```

Manufacturer Name: This field specifies the name of the manufacturer or vendor of a product. It is recommended to set the manufacturer name shown on the device. See section F.2.14 for the format of this field.

Model Name: This field specifies the model name or number of a product.

Serial Number: This field specifies the serial number of a product.

Version Number: This field specifies the version number of a product.

```
Example: Camera product details

<MANUFACTURER>
  <ORG_NAME>Acme</ORG_NAME>
</MANUFACTURER>
<MODEL>Model 1000</MODEL>
<SERIAL>2941554</SERIAL>
```

F.3 Defined Attributes

F.3.1 Language Attribute

The field is formatted according to RFC 1766 [6]. When a metadata field has a Language Attribute, it specifies the language in which the metadata is stored. English (e.g. "en") is assumed where the language is not specified.

Where a field specifies a Language Attribute, and also sub-fields, the language of the sub-fields is the same as the enclosing field unless the Language attribute is specified separately within the sub-field.

```
Schema Definition:

<xsd:attribute name="xml:lang" type="xsd:language"/>
```

```
Example:

<COMMENT xml:lang="fr">
  Bon appel!
</COMMENT>
```

F.3.2 Timestamp Attribute

When a metadata field contains a Timestamp Attribute, it specifies the time that the metadata was generated. See Section 5.6 for details of the use of this attribute.

Where a field specifies a Timestamp Attribute, and also sub-fields, the Timestamp of the sub-fields is the same as the enclosing field unless the Timestamp Attribute is specified separately within the sub-field.

This field contains a string that is ISO 8601 [13] compliant.

```
Schema Definition:

<xsd:attribute name="TIMESTAMP" type="xsd:timeInstant"/>
```

```
Examples:

1968-08-19T18:30
1968-08-19T06:30
1968-08-19T18:30:10+10:00
1968-08-19T18:30:10.52
1968-08-19+10:00
1968-08
```

F.4 Defined Elements

F.4.1 Comment

The Comment field is used to specify extra information to the containing field that cannot be described otherwise within the defined metadata. It is recommended that the Comment field be used as a last resort and only when the other metadata fields are not suitable to store a specific piece of metadata.

The content of this field is intended to store human readable data. Storing non-human readable data can be performed using other metadata extension methods.

This field may contain the sub-field listed below.

```
Schema Definition:

<xsd:element name="COMMENT">
  <xsd:complexType base="dig35:tLangString" derivedBy="extension">
    <xsd:attribute ref="dig35:TIMESTAMP"/>
  </xsd:complexType>
</xsd:element>
```

Annex G: DIG35 XML Document Definition

G.1 Overview

This Annex defines the DIG35 XML document construct and groups the metadata definitions and terms used in Annex A through F. The top-level field that groups all metadata shall be defined as the `METADATA` element.

A DIG35 XML document that contains a `METADATA` element as the root field is defined as a *DIG35 metadata document*. A *DIG35 metadata document* may either be an independent XML document, which may contain other XML fragments with more than one XML namespace, or be a fragment itself and thus would be a sub-tree of a parent XML document. The former type is called a *stand-alone DIG35 metadata document* where the root element starts with the `METADATA` element. The latter type is called a *fragmented DIG35 metadata document*. In both cases, it is expected that DIG35 metadata elements and other XML elements can be distinguished by each XML namespace.

G.2 Structure

A DIG35 XML document is designed to describe metadata for either a single image or a collection of images. Single image metadata would contain, for example, the camera capture information (e.g. date, time, shutter speed, etc.) or a description of the objects (e.g. people and things) for that image.

Metadata for a collection of images would be information that ties the images together, as a group, such that the collection itself comprises a semantic meaning. Such metadata would be, for example, a group of images taken at an event (e.g. a wedding or a birthday party). This could be analogous to the selection criteria of an image database search. Note that each image within a collection may also contain individual metadata and can be recorded in the same metadata document as the image collection metadata. The individual metadata would be stored as a sub-tree of the image collection metadata document.

Therefore, as can be inferred by these examples, the top-level element (i.e. the `METADATA` element) may also be nested. Metadata for collections may contain multiple metadata relative to a single image or even other metadata for a different collection of images and thereby constructing a tree structure. However, metadata for a single image may not contain other image metadata and is then considered as a leaf node.

The `METADATA` element allows each metadata block to be grouped as a single XML document. While the `METADATA` element is considered the root element for a DIG35 metadata document, sub-fields may be stored in separate XML documents and independently be associated with an image.

G.3 Definition

This field is used to describe either a single image, or a collection of images. Note also that this field is used to store previous versions of metadata (and history) within a History field.

This field may contain the sub-fields listed below.

```
Schema Definition:

<xsd:element name="METADATA">
  <xsd:complexType>
    <xsd:element ref="dig35:BASIC_IMAGE_PARAM"                 minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="dig35:IMAGE_CREATION"                    minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="dig35:CONTENT_DESCRIPTION"               minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="dig35:HISTORY"                           minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="dig35:IPR"                               minOccurs="0"  maxOccurs="1"/>
    <xsd:element ref="METADATA"                                minOccurs="0"  maxOccurs="unbounded"/>

    <xsd:attribute name="TYPE"                                 use="default"  value="Single">
      <xsd:simpleType base="xsd:string">
        <xsd:enumeration value="Single"/>
        <xsd:enumeration value="Collection"/>
      </xsd:simpleType>
    </xsd:attribute>

    <xsd:attribute ref="dig35:TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"                              use="default"  value="en"/>
  </xsd:complexType>
</xsd:element>
```

Basic Image Parameters: Described in Annex A: of this document.

Creation Information: Described in Annex B: of this document.

Content Description: Described in Annex C: of this document.

History: Described in Annex D: of this document.

Intellectual Property Rights: Described in Annex E: of this document.

Sub-Metadata: The Metadata field may contain one or more optional Sub-Metadata fields. In addition, a Sub-Metadata field may in turn contain Sub-Metadata. Where there is nested metadata describing a hierarchy of images, then any metadata described at the top levels applies to all nested fields.

Metadata Type: This field specifies the type of metadata the Metadata field is describing. Valid values are listed in Table G-1.

Table G-1: Metadata Type values

Value	Meaning
`Single`	The Metadata field describes a single image. Single type metadata shall not include a sub-metadata field. This is the default value.
`Collection`	The Metadata field describes a collection of images. At least one sub-metadata field that is a single type metadata shall exist. Collection type metadata may be nested as one or more sub-metadata fields.

G.4 Examples

G.4.1 Stand-alone DIG35 Metadata Document

The following example shows a 'stand-alone DIG35 metadata document' for a single image.

```
<?xml version="1.0"?>
<!DOCTYPE METADATA SYSTEM "-//DIG//DTD DIG35 1.0//EN"
                                    "http://www.digitalimaging.org/dig35/1.0/xml/dig35.dtd">

<METADATA TYPE="Single" xmlns="http://www.digitalimaging.org/dig35/1.0/xml">
    <!--
       - The DIG35 XML document; each DIG35 element with no namespace prefix
      -->
</METADATA>
```

The next example shows metadata for a collection of images with two images having its own metadata as well.

```
<?xml version="1.0"?>
<!DOCTYPE METADATA SYSTEM "-//DIG//DTD DIG35 1.0//EN"
                                   " http://www.digitalimaging.org/dig35/1.0/xml/dig35.dtd">


<METADATA TYPE="Collection">
    <IMAGE_CREATION>
        <!--
           - Metadata on how all images were created
          -->
    </IMAGE_CREATION>
    <CONTENT_DESCRIPTION>
        <!—-
           - Metadata on the content of all the images. For example, the event name
           - (e.g. Joe and Mary's wedding)
          -->
    </CONTENT_DESCRIPTION>

    <!--
       -  Metadata for individual images
       -  Metadata for joe.jpg
      -->
    <METADATA TYPE="Single">
        <BASIC_IMAGE_PARAM>
          <BASIC_IMAGE_INFO>
            <FILE_FORMAT>
               <FILE_NAME>joe.jpg</FILE_NAME>
            </FILE_FORMAT>
          </BASIC_IMAGE_INFO>
        </BASIC_IMAGE_PARAM>
        <!--
           - Other metadata on joe.jpg.
          -->
    </METADATA>

    <!--
       -  Metadata for mary.jpg
      -->
    <METADATA TYPE="Single">
        <BASIC_IMAGE_PARAM>
          <BASIC_IMAGE_INFO>
            <FILE_FORMAT>
               <FILE_NAME>mary.jpg</FILE_NAME>
            </FILE_FORMAT>
          </BASIC_IMAGE_INFO>
        </BASIC_IMAGE_PARAM>
        <!--
           - Other metadata on mary.jpg "Mary with college friends"
          -->
    </METADATA>
    <!--
       - Metadata for other images
      -->
</METADATA>
```

G.4.2 Fragmented DIG35 Metadata Document

A fragmented DIG35 metadata document may be embedded within other XML documents. The following is an example that shows a DIG35 metadata document within a SVG document.

```
<?xml version="1.0" standalone="yes"?>

<svg xmlns="http://www.w3.org/2000/svg">
    <metadata>
        <dig35:METADATA xmlns:dig35="http://www.digitalimaging.org/dig35/1.0/xml">
            <!-- The DIG35 XML section, each DIG35 element with namespace prefix -->
        </dig35:METADATA>
    <metadata>
</svg>
```

Annex H: DIG35 XML Schema Collection

NOTE: Because the W3C has not yet completed their efforts on XML Schema, this Annex should be considered *informative*. The Schema as presented follows the current guidelines for XML Schema but may not work correctly with all XML Schema tools available today. Re-ordering the schema may resolve difficulties encountered with some XML Schema tools.

The schema specified here conforms to XML Schema working draft dated April 7, 2000, however, is subject to change as the W3C completes their work on the XML Schema specification. Refer to the website http://www.digitalimaging.org for the latest information on this schema.

```
<!--
   -  This is the XML Schema for DIG35 Metadata Version 1.0 (20000830).
   -
   -  The specification for DIG35 that corresponds to this XML Schema
   -  can be found on the Digital Imaging Group website at:
   -
   -    http://www.digitalimaging.org/
   -
   -  Copyright (c) 2000 Digital Imaging Group, All Rights Reserved.
   -
   -    Namespace:
   -      http://www.digitalimaging.org/dig35/1.0/xml
   -
   -    Public identifier:
   -      PUBLIC "-//DIG//XSD DIG35 1.0//EN"
   -
   -    URI for the XML Schema:
   -      http://www.digitalimaging.org/dig35/1.0/xml/dig35.xsd
  -->
<xsd:schema xmlns="http://www.digitalimaging.org/dig35/1.0/xml"
            xmlns:xml="http://www.w3.org/XML/1998/namespace"
            xmlns:xsd="http://www.w3.org/1999/XMLSchema"
            xmlns:dig35="http://www.digitalimaging.org/dig35/1.0/xml"
            targetNamespace="http://www.digitalimaging.org/dig35/1.0/xml"
            elementFormDefault="qualified">



<!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - See section Annex G: DIG35 XML Document Definition
  -->
<xsd:element name="METADATA">
  <xsd:complexType>
    <xsd:element ref="BASIC_IMAGE_PARAM"                         minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="IMAGE_CREATION"                            minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="CONTENT_DESCRIPTION"                       minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="HISTORY"                                   minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="IPR"                                       minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="METADATA"                                  minOccurs="0" maxOccurs="unbounded"/>

    <xsd:attribute name="TYPE"                                   use="default" value="Single">
      <xsd:simpleType base="xsd:string">
        <xsd:enumeration value="Single"/>
        <xsd:enumeration value="Collection"/>
      </xsd:simpleType>
    </xsd:attribute>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute name="xml:lang" type="xsd:language"           use="default" value="en"/>
```

```
 </xsd:complexType>
</xsd:element>



<!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - See section Annex F:  Fundamental Metadata Types and Fields
  -->

<!--
   - See section F.2.1  Non-negative Double Type
  -->
<xsd:simpleType name="tNonNegativeDouble" base="xsd:double">
    <xsd:minInclusive value="0"/>
</xsd:simpleType>

<!--
   - See section F.2.2  Rational Type
  -->
<xsd:simpleType name="tRational" base="xsd:string">
    <xsd:pattern value="(\-|\+)?[0-9]+/[0-9]+"/>
</xsd:simpleType>

<!--
   - See section F.2.3  String Including Language Type
  -->
<xsd:complexType name="tLangString" base="xsd:string" derivedBy="extension">
  <xsd:attribute ref="xml:lang"/>
</xsd:complexType>


<!--
   - See sections F.2.4  Degree Type
   -                    F.2.5  Half Degree Type
  -->
<xsd:simpleType name="tDegree" base="xsd:double">
    <xsd:minExclusive value="-180"/>
    <xsd:maxInclusive value="180"/>
</xsd:simpleType>

<xsd:simpleType name="tHalfDegree" base="xsd:double">
    <xsd:minExclusive value="-90"/>
    <xsd:maxInclusive value="90"/>
</xsd:simpleType>


<!--
   - See sections F.2.6  Double Size Type
   -                    F.2.7  Integer Size Type
  -->
<xsd:complexType name="tDoubleSize">
  <xsd:element name="WIDTH"          type="tNonNegativeDouble"/>
  <xsd:element name="HEIGHT"         type="tNonNegativeDouble"/>
</xsd:complexType>

<xsd:complexType name="tIntSize">
  <xsd:element name="WIDTH"          type="xsd:positiveInteger"/>
  <xsd:element name="HEIGHT"         type="xsd:positiveInteger"/>
</xsd:complexType>


<!--
   - See section F.2.8  DateTime Type
  -->
<xsd:complexType name="tDateTime">
  <xsd:choice                                                   minOccurs="0" maxOccurs="1">
    <xsd:element name="EXACT"        type="xsd:timeInstant"/>
    <xsd:element name="DATE"         type="xsd:date"/>
    <xsd:sequence>
      <xsd:element name="MONTH"      type="tRecurringMonth"
```

```
            minOccurs="0" maxOccurs="1"/>
      <xsd:element name="YEAR"        type="xsd:year"            minOccurs="0" maxOccurs="1"/>
      <xsd:element name="CENTURY"     type="xsd:century"

            minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:choice>

  <xsd:element name="WEEK_DAY"        type="dig35:tLangString"   minOccurs="0" maxOccurs="1"/>
  <xsd:element name="SEASON"          type="dig35:tLangString"   minOccurs="0" maxOccurs="1"/>

  <xsd:element ref="COMMENT"                                     minOccurs="0" maxOccurs="1"/>

  <xsd:attribute ref="TIMESTAMP"/>
  <xsd:attribute ref="xml:lang"/>
</xsd:complexType>

<xsd:simpleType name="tRecurringMonth" base="xsd:positiveInteger">
  <xsd:minInclusive value="1"/>
  <xsd:maxInclusive value="12"/>
</xsd:simpleType>


<!--
   - See section F.2.9  Address Type
  -->
<xsd:complexType name="tAddress">
  <xsd:element name="ADDR_NAME"       type="tLangString"         minOccurs="0" maxOccurs="1"/>
  <xsd:element ref="ADDR_COMP"                                   minOccurs="0" maxOccurs="unbounded"/>
  <xsd:choice                                                    minOccurs="0" maxOccurs="1">
    <xsd:element name="ZIPCODE"       type="xsd:string"/>
    <xsd:element name="POSTCODE"      type="xsd:string"/>
  </xsd:choice>
  <xsd:element name="COUNTRY"         type="tLangString"         minOccurs="0" maxOccurs="1"/>

  <xsd:attribute name="TYPE"          type="xsd:string"/>
  <xsd:attribute ref="TIMESTAMP"/>
  <xsd:attribute ref="xml:lang"/>
</xsd:complexType>

<xsd:element name="ADDR_COMP">
  <xsd:complexType base="tLangString" derivedBy="extension">
    <xsd:attribute name="TYPE"        type="xsd:string"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section F.2.10  Phone Number Type
  -->
<xsd:complexType name="tPhone">
  <xsd:element name="COUNTRY_CODE"    type="xsd:string"          minOccurs="0" maxOccurs="1"/>
  <xsd:element name="AREA"            type="xsd:string"          minOccurs="0" maxOccurs="1"/>
  <xsd:element name="LOCAL"           type="xsd:string"          minOccurs="0" maxOccurs="1"/>
  <xsd:element name="EXTENSION"       type="xsd:string"          minOccurs="0" maxOccurs="1"/>

  <xsd:attribute name="TYPE"          type="xsd:string"/>
  <xsd:attribute ref="TIMESTAMP"/>
</xsd:complexType>


<!--
   - See section F.2.11  Email Address Type
  -->
<xsd:complexType name="tEmail" base="dig35:tLangString" derivedBy="extension">
  <xsd:attribute name="TYPE"          type="xsd:string"/>
  <xsd:attribute ref="TIMESTAMP"/>
</xsd:complexType>
```

```
<!--
   - See section F.2.12  Web Address Type
  -->
<xsd:complexType name="tWeb" base="dig35:tLangString" derivedBy="extension">
  <xsd:attribute name="TYPE"          type="xsd:string"/>
  <xsd:attribute ref="TIMESTAMP"/>
</xsd:complexType>


<!--
   - See section F.2.13  Person Type
  -->
<xsd:complexType name="tPerson">
  <xsd:element name="NAME_TITLE"      type="tLangString"         minOccurs="0" maxOccurs="1"/>
  <xsd:element ref="PERSON_NAME"                                 minOccurs="0" maxOccurs="unbounded"/>
  <xsd:element name="NICKNAME"        type="dig35:tLangString"   minOccurs="0" maxOccurs="unbounded"/>
  <xsd:element name="JOB_TITLE"       type="dig35:tLangString"   minOccurs="0" maxOccurs="1"/>
  <xsd:choice                                                    minOccurs="0" maxOccurs="1">
    <xsd:element name="PERSON_ORG"    type="tOrganization"/>
    <xsd:element name="ORG_REF"       type="xsd:string"/>
  </xsd:choice>
  <xsd:element name="ADDRESS"         type="tAddress"            minOccurs="0" maxOccurs="unbounded"/>
  <xsd:element name="PHONE"           type="tPhone"              minOccurs="0" maxOccurs="unbounded"/>
  <xsd:element name="EMAIL"           type="tEmail"              minOccurs="0" maxOccurs="unbounded"/>
  <xsd:element name="WEB"             type="tWeb"                minOccurs="0" maxOccurs="unbounded"/>
  <xsd:element name="BIRTH_DATE"      type="xsd:date"            minOccurs="0" maxOccurs="1"/>
  <xsd:element name="AGE"             type="xsd:timeDuration"    minOccurs="0" maxOccurs="1"/>
  <xsd:element ref="COMMENT"                                     minOccurs="0" maxOccurs="1"/>

  <xsd:attribute name="ID"            type="xsd:string"/>
  <xsd:attribute ref="TIMESTAMP"/>
  <xsd:attribute ref="xml:lang"/>
</xsd:complexType>

<xsd:element name="PERSON_NAME">
  <xsd:complexType>
    <xsd:element ref="NAME_COMP"                                                maxOccurs="unbounded"/>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="NAME_COMP">
  <xsd:complexType base="xsd:string" derivedBy="extension">
    <xsd:attribute name="TYPE" use="default" value="Given">
      <xsd:simpleType base="xsd:string">
        <xsd:enumeration value="Prefix"/>
        <xsd:enumeration value="Given"/>
        <xsd:enumeration value="Family"/>
        <xsd:enumeration value="Suffix"/>
        <xsd:enumeration value="Maiden"/>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>


<!--
   - See section F.2.14  Organization Type
  -->
<xsd:complexType name="tOrganization">
  <xsd:element name="ORG_NAME"        type="tLangString"         minOccurs="0" maxOccurs="1"/>
  <xsd:element name="ADDRESS"         type="tAddress"            minOccurs="0" maxOccurs="unbounded"/>
  <xsd:element name="PHONE"           type="tPhone"              minOccurs="0" maxOccurs="unbounded"/>
  <xsd:element name="EMAIL"           type="tEmail"              minOccurs="0" maxOccurs="unbounded"/>
  <xsd:element name="WEB"             type="tWeb"                minOccurs="0" maxOccurs="unbounded"/>
  <xsd:element ref="COMMENT"                                     minOccurs="0" maxOccurs="1"/>
  <xsd:attribute name="ID"            type="xsd:string"/>
  <xsd:attribute ref="TIMESTAMP"/>
  <xsd:attribute ref="xml:lang"/>
</xsd:complexType>


<!--
   - See section F.2.15  Location Type
  -->
<xsd:complexType name="tLocation">
  <xsd:element ref="COORD_LOC"                                  minOccurs="0" maxOccurs="1"/>
  <xsd:element name="ADDRESS"           type="tAddress"          minOccurs="0" maxOccurs="1"/>
  <xsd:element ref="GPS"                                        minOccurs="0" maxOccurs="1"/>
  <xsd:element ref="COMMENT"                                    minOccurs="0" maxOccurs="1"/>

  <xsd:attribute ref="TIMESTAMP"/>
  <xsd:attribute ref="xml:lang"/>
</xsd:complexType>


<!--
   - See section F.2.15.1  Coordinate Location
  -->
<xsd:element name="COORD_LOC">
  <xsd:complexType>
    <xsd:element name="LONGITUDE"     type="tDegree"             minOccurs="0" maxOccurs="1"/>
    <xsd:element name="LATITUDE"      type="tHalfDegree"         minOccurs="0" maxOccurs="1"/>
    <xsd:element name="ALTITUDE"      type="xsd:double"          minOccurs="0" maxOccurs="1"/>

  <xsd:attribute ref="TIMESTAMP"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section F.2.15.2  Raw GPS Information
  -->
<xsd:element name="GPS">
  <xsd:complexType>
    <xsd:element name="GPS_LAT_REF"                                      minOccurs="0" maxOccurs="1">
      <xsd:simpleType base="xsd:string">
        <xsd:enumeration value="N"/>
        <xsd:enumeration value="S"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="GPS_LATITUDE"   type="tDms"                       minOccurs="0" maxOccurs="1"/>
    <xsd:element name="GPS_LONG_REF"                                     minOccurs="0" maxOccurs="1">
      <xsd:simpleType base="xsd:string">
        <xsd:enumeration value="E"/>
        <xsd:enumeration value="W"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="GPS_LONGITUDE"  type="tDms"                       minOccurs="0" maxOccurs="1"/>
    <xsd:element name="GPS_ALTITUDE"   type="tNonNegativeDouble"         minOccurs="0" maxOccurs="1"/>
    <xsd:element name="GPS_TIME"       type="xsd:timeInstant"            minOccurs="0" maxOccurs="1"/>
    <xsd:element name="GPS_SATELLITES" type="xsd:string"                 minOccurs="0" maxOccurs="1"/>
    <xsd:element name="GPS_STATUS"                                       minOccurs="0" maxOccurs="1">
      <xsd:simpleType base="xsd:string">
        <xsd:enumeration value="A"/>
        <xsd:enumeration value="V"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="GPS_MEASURE_MODE"                                 minOccurs="0" maxOccurs="1">
      <xsd:simpleType base="xsd:positiveInteger">
        <xsd:minExclusive value="2"/>
        <xsd:maxInclusive value="3"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="GPS_DOP"        type="tNonNegativeDouble"         minOccurs="0" maxOccurs="1"/>
    <xsd:element name="GPS_SPEED_REF"                                    minOccurs="0" maxOccurs="1">
```

```
        <xsd:simpleType base="xsd:string">
          <xsd:enumeration value="K"/>
          <xsd:enumeration value="N"/>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="GPS_SPEED"      type="tNonNegativeDouble"         minOccurs="0" maxOccurs="1"/>
      <xsd:element name="GPS_TRACK_REF"                                    minOccurs="0" maxOccurs="1">
        <xsd:simpleType base="xsd:string">
          <xsd:enumeration value="T"/>
          <xsd:enumeration value="M"/>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="GPS_TRACK"      type="tNonNegativeDouble"         minOccurs="0" maxOccurs="1"/>
      <xsd:element name="GPS_IMAGE_DIR_REF"                                minOccurs="0" maxOccurs="1">
        <xsd:simpleType base="xsd:string">
          <xsd:enumeration value="T"/>
          <xsd:enumeration value="M"/>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="GPS_IMAGE_DIR"  type="tNonNegativeDouble"         minOccurs="0" maxOccurs="1"/>
      <xsd:element name="GPS_MAP_DATUM"  type="xsd:string"                 minOccurs="0" maxOccurs="1"/>
      <xsd:element name="GPS_DEST_LAT_REF"                                 minOccurs="0" maxOccurs="1">
        <xsd:simpleType base="xsd:string">
          <xsd:enumeration value="N"/>
          <xsd:enumeration value="S"/>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="GPS_DEST_LATITUDE"      type="tDms"               minOccurs="0" maxOccurs="1"/>
      <xsd:element name="GPS_DEST_LONG_REF"                                minOccurs="0" maxOccurs="1">
        <xsd:simpleType base="xsd:string">
          <xsd:enumeration value="E"/>
          <xsd:enumeration value="W"/>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="GPS_DEST_LONGITUDE"     type="tDms"               minOccurs="0" maxOccurs="1"/>
      <xsd:element name="GPS_DEST_BEARING_REF"                             minOccurs="0" maxOccurs="1">
        <xsd:simpleType base="xsd:string">
          <xsd:enumeration value="T"/>
          <xsd:enumeration value="M"/>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="GPS_DEST_BEARING" type="tNonNegativeDouble"       minOccurs="0" maxOccurs="1"/>
      <xsd:element name="GPS_DEST_DISTANCE_REF"                            minOccurs="0" maxOccurs="1">
        <xsd:simpleType base="xsd:string">
          <xsd:enumeration value="K"/>
          <xsd:enumeration value="N"/>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="GPS_DEST_DISTANCE" type="tNonNegativeDouble"      minOccurs="0" maxOccurs="1"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:complexType name="tDms">
    <xsd:element name="D" type="xsd:nonNegativeInteger"/>
    <xsd:element name="M" type="xsd:nonNegativeInteger"/>
    <xsd:element name="S" type="tNonNegativeDouble"                        minOccurs="0" maxOccurs="1"/>
  </xsd:complexType>


  <!--
     - See section F.2.16  Direction Type
    -->
  <xsd:complexType name="tDirection">
    <xsd:element name="YAW"   type="tDegree"                               minOccurs="0" maxOccurs="1"/>
    <xsd:element name="PITCH" type="tHalfDegree"                           minOccurs="0" maxOccurs="1"/>
    <xsd:element name="ROLL"  type="tDegree"                               minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="COMMENT"                                             minOccurs="0" maxOccurs="1"/>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
```

```
</xsd:complexType>


<!--
   - See section F.2.17  Position Type
  -->
<xsd:complexType name="tPosition">
  <xsd:choice                                                      minOccurs="0" maxOccurs="1">
    <xsd:element name="POINT"         type="tPoint"/>
    <xsd:element name="RECT"          type="tRect"/>
    <xsd:sequence>
      <xsd:element name="RECT"        type="tRect"/>
      <xsd:element name="REGION"      type="tRegion"/>
    </xsd:sequence>
  </xsd:choice>
  <xsd:element ref="COMMENT"                                       minOccurs="0" maxOccurs="1"/>

  <xsd:attribute ref="TIMESTAMP"/>
  <xsd:attribute ref="xml:lang"/>
</xsd:complexType>


<!--
   - See section F.2.17.1  Point Type
  -->
<xsd:complexType name="tPoint">
  <xsd:element name="X"               type="tNonNegativeDouble"/>
  <xsd:element name="Y"               type="tNonNegativeDouble"/>
</xsd:complexType>


<!--
   - See section F.2.17.2  Rectangle Type
  -->
<xsd:complexType name="tRect" base="tPoint" derivedBy="extension">
  <xsd:element name="WIDTH"           type="tNonNegativeDouble"/>
  <xsd:element name="HEIGHT"          type="tNonNegativeDouble"/>
</xsd:complexType>


<!--
   - See section F.2.17.3  Region Type
  -->
<xsd:complexType name="tRegion">
  <xsd:element name="POINT"           type="tPoint"/>
  <xsd:choice                                              minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="POINT"         type="tPoint"/>
    <xsd:element ref="SPLINE"/>
  </xsd:choice>
</xsd:complexType>

<xsd:element name="SPLINE">
  <xsd:complexType>
    <xsd:element name="X1"            type="tNonNegativeDouble"/>
    <xsd:element name="Y1"            type="tNonNegativeDouble"/>
    <xsd:element name="X2"            type="tNonNegativeDouble"/>
    <xsd:element name="Y2"            type="tNonNegativeDouble"/>
    <xsd:element name="X"             type="tNonNegativeDouble"/>
    <xsd:element name="Y"             type="tNonNegativeDouble"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section F.2.18  Product Details Type
  -->
<xsd:complexType name="tProductDetails">
  <xsd:element name="MANUFACTURER"    type="tOrganization"       minOccurs="0" maxOccurs="1"/>
  <xsd:element name="MODEL"           type="xsd:string"          minOccurs="0" maxOccurs="1"/>
  <xsd:element name="SERIAL"          type="xsd:string"          minOccurs="0" maxOccurs="1"/>
  <xsd:element name="VERSION"         type="xsd:string"          minOccurs="0" maxOccurs="1"/>

  <xsd:attribute ref="TIMESTAMP"/>
  <xsd:attribute ref="xml:lang"/>
</xsd:complexType>


<!--
   - See section F.3.1  Language Attribute
  -->
<xsd:attribute name="xml:lang" type="xsd:language"/>


<!--
   - See section F.3.2  Timestamp Attribute
  -->
<xsd:attribute name="TIMESTAMP" type="xsd:timeInstant"/>


<!--
   - See section F.4.1  Comment
  -->
<xsd:element name="COMMENT">
  <xsd:complexType base="dig35:tLangString"    derivedBy="extension">
    <xsd:attribute ref="TIMESTAMP"/>
  </xsd:complexType>
</xsd:element>



<!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - See section Annex A:  Basic Image Parameter Metadata
  -->
<xsd:element name="BASIC_IMAGE_PARAM">
  <xsd:complexType>
    <xsd:element ref="BASIC_IMAGE_INFO"                          minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="PREF_PRESENTATION_PARAM"                   minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="COLOR_INFO"                                minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="COMPONENT_INFO"                            minOccurs="0" maxOccurs="1"/>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section A.3.1  Basic Image Information
  -->
<xsd:element name="BASIC_IMAGE_INFO">
  <xsd:complexType>
    <xsd:element ref="FILE_FORMAT"                               minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="IMAGE_ID"                                  minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="IMAGE_SIZE"                                minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="COMPRESSION"                               minOccurs="0" maxOccurs="1"/>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section A.3.1.1  File and Format
  -->
<xsd:element name="FILE_FORMAT">
  <xsd:complexType>
    <xsd:element name="FILE_NAME"     type="xsd:uriReference"    minOccurs="0" maxOccurs="1"/>
    <xsd:element name="FORMAT_TYPE"   type="xsd:string"          minOccurs="0" maxOccurs="1"/>
    <xsd:element name="MIME_TYPE"     type="xsd:string"          minOccurs="0" maxOccurs="1"/>
    <xsd:element name="VERSION"       type="xsd:string"          minOccurs="0" maxOccurs="1"/>

  </xsd:complexType>
</xsd:element>


<!--
   - See section A.3.1.2  Image Identifier
  -->
<xsd:element name="IMAGE_ID">
  <xsd:complexType>
    <xsd:element name="UID"           type="xsd:string"          minOccurs="0" maxOccurs="1"/>
    <xsd:element name="ID_TYPE"       type="xsd:uriReference"    minOccurs="0" maxOccurs="1"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section A.3.1.3  Image Size
  -->
<xsd:element name="IMAGE_SIZE"  type="tIntSize"/>


<!--
   - See section A.3.1.4  Compression Method
  -->
<xsd:element name="COMPRESSION" type="xsd:string"/>


<!--
   - See section A.3.2  Preferred Presentation Parameters
  -->
<xsd:element name="PREF_PRESENTATION_PARAM"    type="tDoubleSize"/>


<!--
   - See section A.3.3  Color Information
  -->
<xsd:element name="COLOR_INFO">
  <xsd:complexType>
    <xsd:element ref="COLORSPACE"                                        maxOccurs="unbounded"/>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section A.3.3.1  Colorspace
  -->
<xsd:element name="COLORSPACE">
  <xsd:complexType>
    <xsd:element name="PROFILE_NAME"  type="tLangString"         minOccurs="0" maxOccurs="1"/>
    <xsd:element name="PROFILE_REF"   type="xsd:uriReference"    minOccurs="0" maxOccurs="1"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section A.3.4  Component Information
  -->
<xsd:element name="COMPONENT_INFO">
  <xsd:complexType>
    <xsd:element name="NUM_COMPONENT" type="xsd:positiveInteger" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="PREMULTIPLIED" type="xsd:boolean"         minOccurs="0" maxOccurs="1"/>
    <xsd:element name="COMPONENTS"    type="xsd:string"          minOccurs="0" maxOccurs="1"/>
    <xsd:element name="COMP_SIZE"     type="xsd:positiveInteger" minOccurs="0" maxOccurs="unbounded"/>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>



<!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - See section Annex B:  Image Creation Metadata
  -->
<xsd:element name="IMAGE_CREATION">
  <xsd:complexType>
    <xsd:element ref="GENERAL_CREATION_INFO"                     minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="CAMERA_CAPTURE"                            minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="SCANNER_CAPTURE"                           minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="CAPTURED_ITEM"                             minOccurs="0" maxOccurs="1"/>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section B.3.1  General Creation Information
  -->
<xsd:element name="GENERAL_CREATION_INFO">
  <xsd:complexType>
    <xsd:element name="CREATION_TIME"          type="xsd:timeInstant"    minOccurs="0" maxOccurs="1"/>
    <xsd:element name="IMAGE_SOURCE"           type="tLangString"        minOccurs="0" maxOccurs="1"/>
    <xsd:element name="SCENE_TYPE"             type="tLangString"        minOccurs="0" maxOccurs="1"/>
    <xsd:element name="IMAGE_CREATOR"          type="tPerson"            minOccurs="0" maxOccurs="1"/>
    <xsd:element name="OPERATOR_ORG"           type="tOrganization"      minOccurs="0" maxOccurs="1"/>
    <xsd:element name="OPERATOR_ID"            type="tLangString"        minOccurs="0" maxOccurs="1"/>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section B.3.2  Camera Capture
  -->
<xsd:element name="CAMERA_CAPTURE">
  <xsd:complexType>
    <xsd:element name="CAMERA_INFO"   type="tProductDetails"     minOccurs="0" maxOccurs="1"/>
    <xsd:element name="SOFTWARE_INFO" type="tProductDetails"     minOccurs="0" maxOccurs="1"/>
    <xsd:element name="LENS_INFO"     type="tProductDetails"     minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="DEVICE_CHARACTER"                          minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="CAMERA_SETTINGS"                           minOccurs="0" maxOccurs="1"/>
    <xsd:element name="ACCESSORY"     type="tProductDetails"     minOccurs="0" maxOccurs="unbounded"/>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section B.3.2.4  Device Characterization
  -->
<xsd:element name="DEVICE_CHARACTER">
  <xsd:complexType>
    <xsd:element ref="SENSOR_TECHNOLOGY"                                 minOccurs="0" maxOccurs="1"/>
    <xsd:element name="FOCAL_PLANE_RES"        type="tIntSize"           minOccurs="0" maxOccurs="1"/>
    <xsd:element name="SPECTRAL_SENSITIVITY"                             minOccurs="0" maxOccurs="1">
      <xsd:complexType>
        <xsd:any processContents="skip"/>
      </xsd:complexType>
    </xsd:element>
```

```
      <xsd:element name="ISO_SATURATION"         type="tNonNegativeDouble" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="ISO_NOISE"              type="tNonNegativeDouble" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="SPATIAL_FREQ_RESPONSE"                            minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="CFA_PATTERN"                                      minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="OECF"                                             minOccurs="0" maxOccurs="1"/>
      <xsd:element name="MIN_F_NUMBER"           type="tNonNegativeDouble" minOccurs="0"  maxOccurs="1"/>

      <xsd:attribute ref="TIMESTAMP"/>
      <xsd:attribute ref="xml:lang"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="SENSOR_TECHNOLOGY">
    <xsd:simpleType base="xsd:string">
      <xsd:enumeration value="One-Chip Color Area"/>
      <xsd:enumeration value="Two-Chip Color Area"/>
      <xsd:enumeration value="Three-Chip Color Area"/>
      <xsd:enumeration value="Color Sequential Area"/>
      <xsd:enumeration value="Trilinear"/>
      <xsd:enumeration value="Color Sequential Linear Sensor"/>
    </xsd:simpleType>
  </xsd:element>

  <xsd:element name="SPATIAL_FREQ_RESPONSE">
    <xsd:complexType>
      <xsd:element ref="SPATIAL_FREQ_VAL"                                         maxOccurs="unbounded"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="SPATIAL_FREQ_VAL">
    <xsd:complexType>
      <xsd:element name="SPATIAL_FREQ"  type="tNonNegativeDouble"/>
      <xsd:element name="HORIZ_SFR"     type="tNonNegativeDouble"/>
      <xsd:element name="VERT_SFR"      type="tNonNegativeDouble"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="CFA_PATTERN">
    <xsd:complexType>
      <xsd:element ref="COLOR_ROW"                                                maxOccurs="unbounded"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="COLOR_ROW">
    <xsd:complexType>
      <xsd:element name="COLOR"                                                   maxOccurs="unbounded">
        <xsd:simpleType base="xsd:string">
          <xsd:enumeration value="Red"/>
          <xsd:enumeration value="Green"/>
          <xsd:enumeration value="Blue"/>
          <xsd:enumeration value="Cyan"/>
          <xsd:enumeration value="Magenta"/>
          <xsd:enumeration value="Yellow"/>
          <xsd:enumeration value="White"/>
        </xsd:simpleType>
      </xsd:element>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="OECF">
    <xsd:complexType>
      <xsd:element ref="LOG_VAL"                                                  maxOccurs="unbounded"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="LOG_VAL">
    <xsd:complexType>
      <xsd:element name="LOG_EXPOSURE"  type="xsd:double"/>
      <xsd:element name="OUTPUT_LEVEL"  type="tNonNegativeDouble"
```

```
            maxOccurs="unbounded"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section B.3.2.5  Camera Capture Settings
  -->
<xsd:element name="CAMERA_SETTINGS">
  <xsd:complexType>
    <xsd:choice                                                          minOccurs="0" maxOccurs="1">
      <xsd:element name="EXP_TIME"             type="tNonNegativeDouble"/>
      <xsd:element name="R_EXP_TIME"           type="tRational"/>
    </xsd:choice>

    <xsd:element name="F_NUMBER"               type="tNonNegativeDouble" minOccurs="0"  maxOccurs="1"/>
    <xsd:element name="EXP_PROGRAM"            type="tLangString"        minOccurs="0" maxOccurs="1"/>
    <xsd:element name="BRIGHTNESS"             type="xsd:double"         minOccurs="0" maxOccurs="1"/>
    <xsd:element name="EXPOSURE_BIAS"          type="xsd:double"         minOccurs="0" maxOccurs="1"/>
    <xsd:element name ="SUBJECT_DISTANCE"      type="tNonNegativeDouble" minOccurs="0" maxOccurs="1"/>
    <xsd:element name ="METERING_MODE"         type="tLangString"        minOccurs="0" maxOccurs="1"/>
    <xsd:element name ="SCENE_ILLUMINANT"      type="tLangString"        minOccurs="0" maxOccurs="1"/>
    <xsd:element name="COLOR_TEMP"             type="tNonNegativeDouble" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="FOCAL_LENGTH"           type="tNonNegativeDouble" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="FLASH"                  type="xsd:boolean"        minOccurs="0" maxOccurs="1"/>
    <xsd:element name="FLASH_ENERGY"           type="tNonNegativeDouble" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="FLASH_RETURN"           type="xsd:boolean"        minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="BACK_LIGHT"                                        minOccurs="0" maxOccurs="1"/>
    <xsd:element name="SUBJECT_POSITION"       type="tPosition"          minOccurs="0" maxOccurs="1"/>
    <xsd:element name="EXPOSURE_INDEX"         type="xsd:double"         minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="AUTO_FOCUS"                                        minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="SPECIAL_EFFECT"                            minOccurs="0"maxOccurs="unbounded"/>
    <xsd:element name="CAMERA_LOCATION"        type="tLocation"          minOccurs="0" maxOccurs="1"/>
    <xsd:element name="ORIENTATION"            type="tDirection"         minOccurs="0" maxOccurs="1"/>
    <xsd:element name="PAR"                    type="tRational"          minOccurs="0" maxOccurs="1"/>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="BACK_LIGHT">
  <xsd:simpleType base="xsd:string">
    <xsd:enumeration value="Front Light"/>
    <xsd:enumeration value="Back Light 1"/>
    <xsd:enumeration value="Back Light 2"/>
  </xsd:simpleType>
</xsd:element>

<xsd:element name="AUTO_FOCUS">
  <xsd:simpleType base="xsd:string">
    <xsd:enumeration value="Auto Focus Used"/>
    <xsd:enumeration value="Auto Focus Interrupted"/>
    <xsd:enumeration value="Near Focused"/>
    <xsd:enumeration value="Soft Focused"/>
    <xsd:enumeration value="Manual"/>
  </xsd:simpleType>
</xsd:element>

<xsd:element name="SPECIAL_EFFECT">
  <xsd:simpleType base="xsd:string">
    <xsd:enumeration value="Colored"/>
    <xsd:enumeration value="Diffusion"/>
    <xsd:enumeration value="Multi-Image"/>
    <xsd:enumeration value="Polarizing"/>
    <xsd:enumeration value="Split-Field"/>
    <xsd:enumeration value="Star"/>
  </xsd:simpleType>
</xsd:element>
```

```
<!--
   - See section B.3.3  Scanner Capture
  -->
<xsd:element name="SCANNER_CAPTURE">
 <xsd:complexType>
    <xsd:element name="SCANNER_INFO"  type="tProductDetails"     minOccurs="0" maxOccurs="1"/>
          <xsd:element name="SOFTWARE_INFO" type="tProductDetails"       minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="SCANNER_SETTINGS"                          minOccurs="0" maxOccurs="1"/>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section B.3.3.3  Scanner Capture Settings
  -->
<xsd:element name="SCANNER_SETTINGS">
  <xsd:complexType>
    <xsd:element name="PIXEL_SIZE"             type="tNonNegativeDouble" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="PHYSICAL_SCAN_RES"      type="tDoubleSize"        minOccurs="0" maxOccurs="1"/>

    <xsd:attribute ref="TIMESTAMP"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section B.3.4  Captured Item
  -->
<xsd:element name="CAPTURED_ITEM">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element ref="REFLECTION_PRINT"/>
      <xsd:element ref="FILM"/>
    </xsd:choice>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section B.3.4.1  Reflection Print
  -->
<xsd:element name="REFLECTION_PRINT">
  <xsd:complexType>
    <xsd:element name="DOCUMENT_SIZE" type="tDoubleSize"
                                                                                           minOccurs="0"
maxOccurs="1"/>
    <xsd:element ref="MEDIUM"                                            minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="RP_TYPE"                                           minOccurs="0" maxOccurs="1"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="MEDIUM">
  <xsd:simpleType base="xsd:string">
    <xsd:enumeration value="Continuous Tone Image"/>
    <xsd:enumeration value="Halftone Image"/>
    <xsd:enumeration value="Line Art"/>
  </xsd:simpleType>
</xsd:element>

<xsd:element name="RP_TYPE">
  <xsd:simpleType base="xsd:string">
    <xsd:enumeration value="B/W Print"/>
    <xsd:enumeration value="Color Print"/>
    <xsd:enumeration value="B/W Document"/>
    <xsd:enumeration value="Color Document"/>
  </xsd:simpleType>
</xsd:element>


<!--
   - See section B.3.4.2  Film
  -->
<xsd:element name="FILM">
  <xsd:complexType>
    <xsd:element name="BRAND"         type="tProductDetails"     minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="CATEGORY"                                  minOccurs="0" maxOccurs="1"/>
    <xsd:element name="FILM_SIZE"     type="tDoubleSize"         minOccurs="0" maxOccurs="1"/>
    <xsd:element name="ROLL_ID"       type="tLangString"         minOccurs="0" maxOccurs="1"/>
    <xsd:element name="FRAME_ID"      type="xsd:positiveInteger" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="FILM_SPEED"    type="xsd:positiveInteger" minOccurs="0" maxOccurs="1"/>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="CATEGORY">
  <xsd:simpleType base="xsd:string">
    <xsd:enumeration value="Negative B/W"/>
    <xsd:enumeration value="Negative Color"/>
    <xsd:enumeration value="Reversal B/W"/>
    <xsd:enumeration value="Reversal Color"/>
    <xsd:enumeration value="Chromagenic"/>
    <xsd:enumeration value="Internegative B/W"/>
    <xsd:enumeration value="Internegative Color"/>
  </xsd:simpleType>
</xsd:element>



<!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - See section Annex C:  Content Description Metadata
  -->
<xsd:element name="CONTENT_DESCRIPTION">
  <xsd:complexType>
    <xsd:element name="GROUP_CAPTION" type="tLangString"         minOccurs="0" maxOccurs="1"/>
    <xsd:element name="CAPTION"       type="tLangString"         minOccurs="0" maxOccurs="1"/>
    <xsd:element name="CAPTURE_TIME"  type="tDateTime"           minOccurs="0" maxOccurs="1"/>
    <xsd:element name="LOCATION"      type="tLocation"           minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="PERSON"                                    minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="THING"                                     minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="ORGANIZATION"                              minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="EVENT"                                     minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="AUDIO"                                     minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="PROPERTY"                                  minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="DICTIONARY"                                minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="COMMENT"                                   minOccurs="0" maxOccurs="1"/>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section C.3.5  Person Description
  -->
<xsd:element name="PERSON">
  <xsd:complexType base="tPerson" derivedBy="extension">
    <xsd:element name="POSITION"      type="tPosition"           minOccurs="0" maxOccurs="1"/>
    <xsd:element name="LOCATION"      type="tLocation"           minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="PROPERTY"                                  minOccurs="0" maxOccurs="unbounded"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section C.3.6  Thing Description
  -->
<xsd:element name="THING">
  <xsd:complexType>
    <xsd:element name="NAME"          type="tLangString"          minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="COMMENT"                                    minOccurs="0" maxOccurs="1"/>
    <xsd:element name="POSITION"      type="tPosition"            minOccurs="0" maxOccurs="1"/>
    <xsd:element name="LOCATION"      type="tLocation"            minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="PROPERTY"                                   minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="THING"                                      minOccurs="0" maxOccurs="unbounded"/>

    <xsd:attribute name="ID"          type="xsd:string"/>
    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section C.3.7  Organization Description
  -->
<xsd:element name="ORGANIZATION">
  <xsd:complexType base="tOrganization" derivedBy="extension">
    <xsd:element name="POSITION" type="tPosition"                 minOccurs="0" maxOccurs="1"/>
    <xsd:element name="LOCATION" type="tLocation"                 minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="PROPERTY"                                   minOccurs="0" maxOccurs="unbounded"/>

  </xsd:complexType>
</xsd:element>


<!--
   - See section C.3.8  Event Description
  -->
<xsd:element name="EVENT">
  <xsd:complexType>
    <xsd:element name="EVENT_TYPE"    type="tLangString"/>
    <xsd:element name="DESCRIPTION"   type="tLangString"          minOccurs="0" maxOccurs="1"/>
    <xsd:element name="LOCATION"      type="tLocation"            minOccurs="0" maxOccurs="1"/>
    <xsd:element name="EVENT_TIME"    type="tDateTime"            minOccurs="0" maxOccurs="1"/>
    <xsd:element name="DURATION"      type="xsd:timeDuration"     minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="COMMENT"                                    minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="PARTICIPANT"                                minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="EVENT_RELATION"                             minOccurs="0" maxOccurs="unbounded"/>
    <xsd:choice                                                   minOccurs="0" maxOccurs="unbounded">
      <xsd:element ref="EVENT"/>
      <xsd:element name="EVENT_REF"   type="xsd:string"/>
    </xsd:choice>

    <xsd:attribute name="ID"          type="xsd:string"/>
    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="PARTICIPANT">
  <xsd:complexType>
      <xsd:element name="ROLE"        type="tLangString"          minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="OBJECT_REF"  type="xsd:string"/>

    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="EVENT_RELATION">
```

```
    <xsd:complexType>
      <xsd:element name="RELATION"      type="tLangString"        minOccurs="0"  maxOccurs="unbounded"/>
      <xsd:element name="EVENT_REF"     type="xsd:string"                        maxOccurs="unbounded"/>
    </xsd:complexType>
  </xsd:element>


  <!--
     - See section C.3.9  Audio
    -->
  <xsd:element name="AUDIO">
    <xsd:complexType>
      <xsd:element name="AUDIO_STREAM"  type="xsd:uriReference"/>
      <xsd:element name="AUDIO_FORMAT"  type="xsd:string"         minOccurs="0" maxOccurs="1"/>
      <xsd:element name="MIME_TYPE"     type="xsd:string"         minOccurs="0" maxOccurs="1"/>
      <xsd:element name="DESCRIPTION"   type="tLangString"        minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="COMMENT"                                  minOccurs="0" maxOccurs="1"/>

      <xsd:attribute ref="TIMESTAMP"/>
      <xsd:attribute ref="xml:lang"/>
    </xsd:complexType>
  </xsd:element>


  <!--
     - See section C.3.10  Property
    -->
  <xsd:element name="PROPERTY">
    <xsd:complexType>
      <xsd:element name="NAME"          type="tLangString"        minOccurs="0" maxOccurs="1"/>
      <xsd:element name="VALUE"         type="tLangString"        minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="COMMENT"                                  minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="PROPERTY"                                 minOccurs="0" maxOccurs="unbounded"/>

      <xsd:attribute name="DICT_REF"    type="xsd:string"/>
      <xsd:attribute ref="TIMESTAMP"/>
      <xsd:attribute ref="xml:lang"/>
    </xsd:complexType>
  </xsd:element>


  <!--
     - See section C.3.11  Dictionary Definition
    -->
  <xsd:element name="DICTIONARY">
    <xsd:complexType>
      <xsd:element name="DICT_NAME"     type="tLangString"        minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="COMMENT"                                  minOccurs="0" maxOccurs="1"/>

      <xsd:attribute name="ID"          type="xsd:string"/>
      <xsd:attribute ref="TIMESTAMP"/>
      <xsd:attribute ref="xml:lang"/>
    </xsd:complexType>
  </xsd:element>



  <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     - See section Annex D:  History Metadata
    -->
  <xsd:element name="HISTORY">
    <xsd:complexType>
      <xsd:element ref="PROCESSING_SUMMARY"                       minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="IMAGE_PROCESSING_HINTS"                   minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="METADATA"                                 minOccurs="0" maxOccurs="unbounded"/>

      <xsd:attribute ref="TIMESTAMP"/>
      <xsd:attribute ref="xml:lang"/>
    </xsd:complexType>
  </xsd:element>
```

```
<!--
   - See section D.3.1  Processing Summary
  -->
<xsd:element name="PROCESSING_SUMMARY">
  <xsd:complexType>
    <xsd:element name="IMG_CREATED"                              minOccurs="0" maxOccurs="1">
      <xsd:complexType content="empty"/>
    </xsd:element>
    <xsd:element name="IMG_CROPPED"                              minOccurs="0" maxOccurs="1">
      <xsd:complexType content="empty"/>
    </xsd:element>
    <xsd:element name="IMG_TRANSFORMED"                          minOccurs="0" maxOccurs="1">
      <xsd:complexType content="empty"/>
    </xsd:element>
    <xsd:element name="IMG_GTC_ADJ"                              minOccurs="0" maxOccurs="1">
      <xsd:complexType content="empty"/>
    </xsd:element>
    <xsd:element name="IMG_STC_ADJ"                              minOccurs="0" maxOccurs="1">
      <xsd:complexType content="empty"/>
    </xsd:element>
    <xsd:element name="IMG_SPATIAL_ADJ"                          minOccurs="0" maxOccurs="1">
      <xsd:complexType content="empty"/>
    </xsd:element>
    <xsd:element name="IMG_EXT_EDITED"                           minOccurs="0" maxOccurs="1">
      <xsd:complexType content="empty"/>
    </xsd:element>
    <xsd:element name="IMG_RETOUCHED"                            minOccurs="0" maxOccurs="1">
      <xsd:complexType content="empty"/>
    </xsd:element>
    <xsd:element name="IMG_COMPOSITED"                           minOccurs="0" maxOccurs="1">
      <xsd:complexType content="empty"/>
    </xsd:element>
    <xsd:element name="IMG_METADATA"                             minOccurs="0" maxOccurs="1">
      <xsd:complexType content="empty"/>
    </xsd:element>

    <xsd:attribute ref="TIMESTAMP"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section D.3.2  Image Processing Hints
  -->
<xsd:element name="IMAGE_PROCESSING_HINTS">
  <xsd:complexType>
    <xsd:choice                                                  minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="IMG_CREATED"          type="tLangString"/>
      <xsd:element name="IMG_CROPPED"          type="tLangString"/>
      <xsd:element name="IMG_TRANSFORMED"      type="tLangString"/>
      <xsd:element name="IMG_GTC_ADJ"          type="tLangString"/>
      <xsd:element name="IMG_STC_ADJ"          type="tLangString"/>
      <xsd:element name="IMG_SPATIAL_ADJ"      type="tLangString"/>
      <xsd:element name="IMG_EXT_EDITED"       type="tLangString"/>
      <xsd:element name="IMG_RETOUCHED"        type="tLangString"/>
      <xsd:element name="IMG_COMPOSITED"       type="tLangString"/>
      <xsd:element name="IMG_METADATA"         type="tLangString"/>
    </xsd:choice>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>



<!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - See section Annex E:  Intellectual Property Rights Metadata
  -->
<xsd:element name="IPR">
  <xsd:complexType>
    <xsd:element ref="IPR_NAMES"                                 minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="IPR_DESCRIPTION"                           minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="IPR_DATES"                                 minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="IPR_EXPLOITATION"                          minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="IPR_IDENTIFICATION"                        minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="IPR_CONTACT_POINT"                         minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="IPR_HISTORY"                               minOccurs="0" maxOccurs="1"/>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section E.3.1  Names
  -->
<xsd:element name="IPR_NAMES">
  <xsd:complexType>
    <xsd:choice
maxOccurs="unbounded">
      <xsd:element ref="IPR_PERSON"/>
      <xsd:element ref="IPR_ORG"/>
      <xsd:element ref="IPR_NAME_REF"/>
    </xsd:choice>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="IPR_PERSON">
  <xsd:complexType base="tPerson" derivedBy="extension">
    <xsd:attribute name="DESCRIPTION" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="IPR_ORG">
  <xsd:complexType base="tOrganization" derivedBy="extension">
    <xsd:attribute name="DESCRIPTION" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="IPR_NAME_REF">
  <xsd:complexType base="xsd:string" derivedBy="extension">
    <xsd:attribute name="DESCRIPTION" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section E.3.2  Description
  -->
<xsd:element name="IPR_DESCRIPTION">
  <xsd:complexType>
    <xsd:element name="IPR_TITLE"     type="tLangString"         minOccurs="0" maxOccurs="1"/>
    <xsd:element name="IPR_LEGEND"    type="tLangString"         minOccurs="0" maxOccurs="1"/>
    <xsd:element name="IPR_CAPTION"   type="tLangString"         minOccurs="0" maxOccurs="1"/>
    <xsd:element name="COPYRIGHT"     type="tLangString"         minOccurs="0" maxOccurs="1"/>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>


<!--
```

```
   - See section E.3.3  Dates
 -->
<xsd:element name="IPR_DATES">
  <xsd:complexType>
    <xsd:element ref="IPR_DATE"                                                 maxOccurs="unbounded"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="IPR_DATE">
  <xsd:complexType base="tDateTime" derivedBy="extension">
    <xsd:attribute name="DESCRIPTION" type="xsd:string"/>
    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section E.3.4  Exploitation
 -->
<xsd:element name="IPR_EXPLOITATION">
  <xsd:complexType>
    <xsd:element name="IPR_PROTECTION" type="xsd:nonNegativeInteger"     minOccurs="0" maxOccurs="1"/>
    <xsd:element name="IPR_USE_RESTRICTION"    type="tLangString"        minOccurs="0" maxOccurs="1"/>
    <xsd:element name="IPR_OBLIGATION"         type="tLangString"        minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="IPR_MGMT_SYS"                                      minOccurs="0" maxOccurs="1"/>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section E.3.4.4  IPR Management System
 -->
<xsd:element name="IPR_MGMT_SYS">
  <xsd:complexType>
    <xsd:element name="IPR_MGMT_TYPE"          type="xsd:string"         minOccurs="0" maxOccurs="1"/>
    <xsd:element name="IPR_MGMT_SYS_ID"        type="xsd:string"         minOccurs="0" maxOccurs="1"/>
    <xsd:element name="IPR_MGMT_SYS_LOCATION"  type="xsd:uriReference"   minOccurs="0" maxOccurs="1"/>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section E.3.5  Identification
 -->
<xsd:element name="IPR_IDENTIFICATION">
  <xsd:complexType>
    <xsd:element ref="IPR_IDENTIFIER"                                    minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="LICENCE_PLATE"                                     minOccurs="0" maxOccurs="1"/>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section E.3.5.1  Generic IPR Identifier
 -->
<xsd:element name="IPR_IDENTIFIER">
  <xsd:complexType>
    <xsd:element name="IPR_ID_MODE"            type="tLangString"        minOccurs="0" maxOccurs="1"/>
    <xsd:element name="IPR_ID"                 type="tLangString"        minOccurs="0" maxOccurs="1"/>
  </xsd:complexType>
```

```
</xsd:element>


<!--
   - See section E.3.5.2  Licence Plate
  -->
<xsd:element name="LICENCE_PLATE">
  <xsd:complexType>
    <xsd:element name="LP_COUNTRY"            type="xsd:string"         minOccurs="0" maxOccurs="1"/>
    <xsd:element name="LP_REG_AUT"            type="xsd:string"         minOccurs="0" maxOccurs="1"/>
    <xsd:element name="LP_REG_NUM"            type="xsd:string"         minOccurs="0" maxOccurs="1"/>
    <xsd:element name="LP_DELIVERY_DATE"      type="xsd:timeInstant"    minOccurs="0" maxOccurs="1"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section E.3.6  Contact Point
  -->
<xsd:element name="IPR_CONTACT_POINT">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element ref="IPR_PERSON"/>
      <xsd:element ref="IPR_ORG"/>
      <xsd:element name="IPR_NAME_REF"         type="xsd:string"/>
    </xsd:choice>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>


<!--
   - See section E.3.7  IPR History
  -->
<xsd:element name="IPR_HISTORY">
  <xsd:complexType>
    <xsd:element ref="IPR"                                            minOccurs="0" maxOccurs="unbounded"/>

    <xsd:attribute ref="TIMESTAMP"/>
    <xsd:attribute ref="xml:lang"/>
  </xsd:complexType>
</xsd:element>

</xsd:schema>
```

Annex I: DIG35 XML DTD

```
<!--
   -  This is the DTD for DIG35 Metadata Version 1.0 (20000830).
   -
   -  The specification for DIG35 that corresponds to this DTD
   -  can be found on the Digital Imaging Group website at:
   -
   -    http://www.digitalimaging.org/
   -
   -  Copyright (c) 2000 Digital Imaging Group, All Rights Reserved.
   -
   -    Namespace:
   -      http://www.digitalimaging.org/dig35/1.0/xml
   -
   -    Public identifier:
   -      PUBLIC "-//DIG//DTD DIG35 1.0//EN"
   -
   -    URI for the DTD:
   -      http://www.digitalimaging.org/dig35/1.0/xml/dig35.dtd
  -->

<!-- Predefined general entities -->
<!ENTITY  lt                                "&#38;#60;">
<!ENTITY  gt                                "&#62;">
<!ENTITY  amp                               "&#38;#38;">
<!ENTITY  apos                              "&#39;">
<!ENTITY  quot                              "&#34;">
<!ENTITY  copy                              "&#169;">

<!--
   -  XML Schema defined types
   -
   -  See: http://www.w3.org/TR/xmlschema-2/
  -->
<!ENTITY % xsd-language-c                   "CDATA">
<!ENTITY % xsd-string-c                     "CDATA">
<!ENTITY % xsd-timeInstant-c                "CDATA">
<!ENTITY % xsd-boolean                      "#PCDATA">
<!ENTITY % xsd-century                      "#PCDATA">
<!ENTITY % xsd-date                         "#PCDATA">
<!ENTITY % xsd-double                       "#PCDATA">
<!ENTITY % xsd-nonNegativeInteger           "#PCDATA">
<!ENTITY % xsd-positiveInteger              "#PCDATA">
<!ENTITY % xsd-string                       "#PCDATA">
<!ENTITY % xsd-timeDuration                 "#PCDATA">
<!ENTITY % xsd-timeInstant                  "#PCDATA">
<!ENTITY % xsd-uriReference                 "#PCDATA">
<!ENTITY % xsd-year                         "#PCDATA">

<!--
   - DIG35 defined types
  -->
<!ENTITY % tDegree                          "#PCDATA">
<!ENTITY % tHalfDegree                      "#PCDATA">
<!ENTITY % tNonNegativeDouble               "#PCDATA">
<!ENTITY % tRational                        "#PCDATA">
<!ENTITY % tRecurringMonth                  "#PCDATA">
<!--
   - tDegree           See section F.2.4  Degree Type
   - tHalfDegree       See section F.2.5  Half Degree Type
   - tNonNegaiveDouble See section F.2.1  Non-negative Double Type
   - tRational         See section F.2.2  Rational Type
   - tRecurringMonth   See section F.2.8  DateTime Type
  -->

<!--
   - Attribute definitions
  -->
<!ENTITY % att-timestamp                     "TIMESTAMP %xsd-timeInstant-c; #IMPLIED">
<!ENTITY % att-lang                          "xml:lang %xsd-language-c; #IMPLIED">
<!ENTITY % att-lang-ts                       "%att-lang; %att-timestamp;">
<!ENTITY % att-lang-ts-id                    "%att-lang-ts; ID %xsd-string-c; #IMPLIED">
<!--
   - att-timestamp     See section F.3.2  Timestamp Attribute
   - att-lang          See section F.3.1  Language Attribute
   - att-lang-ts       Language and timestamp attributes
   - att-lang-ts-id    Language timestamp and id attributes
  -->



<!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - See section Annex G: DIG35 XML Document Definition
  -->
<!ELEMENT METADATA                         ((BASIC_IMAGE_PARAM?,
                                             IMAGE_CREATION?,
                                             CONTENT_DESCRIPTION?,
                                             HISTORY?,
                                             IPR?),
                                            METADATA*)>
<!ATTLIST METADATA                          TYPE (Single | Collection) "Single"
                                            xml:lang %xsd-language-c; "en"
                                            %att-timestamp;>

<!ATTLIST METADATA    xmlns CDATA #FIXED    "http://www.digitalimaging.org/dig35/1.0/xml"


<!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - See section Annex F:  Fundamental Metadata Types and Fields
  -->

<!--
   - See sections F.2.6  Double Size Type
   -          F.2.7  Integer Size Type
   -
   - Note as DTDs do not specify type, the same entity is used for both types.
  -->
<!ENTITY % size                             "(WIDTH, HEIGHT)">


<!--
   - See section F.2.8  DateTime Type
  -->
<!ENTITY % dig35-tDateTime                  "(EXACT | DATE |
                                            (MONTH?, YEAR?, CENTURY?)),
                                             WEEK_DAY?, SEASON?, COMMENT?">


<!ELEMENT EXACT                             (%xsd-timeInstant;)>
<!ELEMENT DATE                              (%xsd-date;)>
<!ELEMENT MONTH                             (%tRecurringMonth;)>
<!ELEMENT YEAR                              (%xsd-year;)>
<!ELEMENT CENTURY                           (%xsd-century;)>

<!ELEMENT WEEK_DAY                          (%xsd-string;)>
<!ATTLIST WEEK_DAY                          %att-lang;>

<!ELEMENT SEASON                            (%xsd-string;)>
<!ATTLIST SEASON                            %att-lang;>


<!--
```

```
   - See section F.2.9  Address Type
  -->
<!ENTITY % dig35-tAddress                  "(ADDR_NAME?, ADDR_COMP*,
                                            (POSTCODE | ZIPCODE)?,
                                            COUNTRY?)">
<!ELEMENT ADDRESS                           %dig35-tAddress;>
<!ATTLIST ADDRESS                           TYPE %xsd-string-c; #IMPLIED
                                            %att-lang-ts;>

<!ELEMENT ADDR_NAME                        (%xsd-string;)>
<!ATTLIST ADDR_NAME                        %att-lang;>

<!ELEMENT ADDR_COMP                        (%xsd-string;)>
<!ATTLIST ADDR_COMP                        TYPE %xsd-string-c; #IMPLIED>

<!ELEMENT POSTCODE                         (%xsd-string;)>
<!ELEMENT ZIPCODE                          (%xsd-string;)>

<!ELEMENT COUNTRY                          (%xsd-string;)>
<!ATTLIST COUNTRY                          %att-lang;>


<!--
   - See section F.2.10  Phone Number Type
  -->
<!ENTITY % dig35-tPhone                     "(COUNTRY_CODE?, AREA?,
                                            LOCAL?, EXTENSION?)">
<!ATTLIST PHONE                             TYPE %xsd-string-c; #IMPLIED
                                            %att-timestamp;>
<!ELEMENT PHONE                            %dig35-tPhone;>

<!ELEMENT COUNTRY_CODE                     (%xsd-string;)>
<!ELEMENT AREA                             (%xsd-string;)>
<!ELEMENT LOCAL                            (%xsd-string;)>
<!ELEMENT EXTENSION                        (%xsd-string;)>


<!--
   - See section F.2.11  Email Address Type
  -->
<!ELEMENT EMAIL                             (%xsd-string;)>
<!ATTLIST EMAIL                             TYPE %xsd-string-c; #IMPLIED>


<!--
   - See section F.2.12  Web Address Type
  -->
<!ELEMENT WEB                               (%xsd-string;)>
<!ATTLIST WEB                               TYPE %xsd-string-c; #IMPLIED>


<!--
   - See section F.2.14  Organization Type
  -->
<!ENTITY % dig35-tOrganization             "(ORG_NAME?,
                                            ADDRESS*, PHONE*, EMAIL*, WEB*,
                                            COMMENT?)">

<!ELEMENT ORG_NAME                          (%xsd-string;)>
<!ATTLIST ORG_NAME                          %att-lang;>


<!--
   - See section F.2.13  Person Type
  -->
<!ENTITY % dig35-tPerson                    "(NAME_TITLE?,
                                             PERSON_NAME*, NICKNAME*,
                                             JOB_TITLE?,
                                             (PERSON_ORG, ORG_REF)?,
                                             ADDRESS*, PHONE*, EMAIL*, WEB*,
                                            BIRTH_DATE?, AGE?,
                                            COMMENT?)">

<!ELEMENT NAME_TITLE                       (%xsd-string;)>
<!ATTLIST NAME_TITLE                       %att-lang;>

<!ELEMENT PERSON_NAME                      (NAME_COMP+)>
<!ATTLIST PERSON_NAME                      %att-lang-ts;>

<!ELEMENT NAME_COMP                        (%xsd-string;)>
<!ATTLIST NAME_COMP                        TYPE (Prefix | Given | Family |
                                                 Suffix | Maiden) "Given">

<!ELEMENT NICKNAME                         (%xsd-string;)>
<!ATTLIST NICKNAME                         %att-lang;>

<!ELEMENT JOB_TITLE                        (%xsd-string;)>
<!ATTLIST JOB_TITLE                        %att-lang;>

<!ELEMENT PERSON_ORG                       %dig35-tOrganization;>
<!ATTLIST PERSON_ORG                       %att-lang-ts-id;>

<!ELEMENT ORG_REF                          (%xsd-string;)>

<!ELEMENT BIRTH_DATE                       (%xsd-date;)>
<!ELEMENT AGE                              (%xsd-timeDuration;)>


<!--
   - See section F.2.15  Location Type
  -->
<!ENTITY % dig35-tLocation                "(COORD_LOC?, ADDRESS?,
                                           GPS?, COMMENT?)">
<!ELEMENT LOCATION                         %dig35-tLocation;>
<!ATTLIST LOCATION                         %att-lang-ts;>


<!--
   - See section F.2.15.1  Coordinate Location
  -->
<!ELEMENT COORD_LOC                       (LONGITUDE?, LATITUDE?, ALTITUDE?)>
<!ATTLIST COORD_LOC                        %att-timestamp;>

<!ELEMENT LONGITUDE                        (%tDegree;)>
<!ELEMENT LATITUDE                         (%tHalfDegree;)>
<!ELEMENT ALTITUDE                         (%xsd-double;)>


<!--
   - See section F.2.15.2  Raw GPS Information
  -->
<!ELEMENT GPS                             (GPS_LAT_REF?, GPS_LATITUDE?,
                                           GPS_LONG_REF?, GPS_LONGITUDE?,
                                           GPS_ALTITUDE?, GPS_TIME?,
                                           GPS_SATELLITES?, GPS_STATUS?,
                                           GPS_MEASURE_MODE?, GPS_DOP?,
                                           GPS_SPEED_REF?, GPS_SPEED?,
                                           GPS_TRACK_REF?, GPS_TRACK?,
                                           GPS_IMAGE_DIR_REF?, GPS_IMAGE_DIR?,
                                           GPS_MAP_DATUM?,
                                           GPS_DEST_LAT_REF?,
                                           GPS_DEST_LATITUDE?,
                                           GPS_DEST_LONG_REF?,
                                           GPS_DEST_LONGITUDE?,
                                           GPS_DEST_BEARING_REF?,
                                           GPS_DEST_BEARING?,
                                           GPS_DEST_DISTANCE_REF?,
                                           GPS_DEST_DISTANCE?)>

<!ELEMENT GPS_LAT_REF                     (%xsd-string;)>
```

```
<!ELEMENT GPS_LATITUDE                      (D, M, S?)>
<!ELEMENT GPS_LONG_REF                      (%xsd-string;)>
<!ELEMENT GPS_LONGITUDE                     (D, M, S?)>
<!ELEMENT GPS_ALTITUDE                      (%tNonNegativeDouble;)>
<!ELEMENT GPS_TIME                          (%xsd-timeInstant;)>
<!ELEMENT GPS_SATELLITES                    (%xsd-string;)>
<!ELEMENT GPS_STATUS                        (%xsd-string;)>
<!ELEMENT GPS_MEASURE_MODE                  (%xsd-positiveInteger;)>
<!ELEMENT GPS_DOP                           (%tNonNegativeDouble;)>
<!ELEMENT GPS_SPEED_REF                     (%xsd-string;)>
<!ELEMENT GPS_SPEED                         (%tNonNegativeDouble;)>
<!ELEMENT GPS_TRACK_REF                     (%xsd-string;)>
<!ELEMENT GPS_TRACK                         (%tNonNegativeDouble;)>
<!ELEMENT GPS_IMAGE_DIR_REF                 (%xsd-string;)>
<!ELEMENT GPS_IMAGE_DIR                     (%tNonNegativeDouble;)>
<!ELEMENT GPS_MAP_DATUM                     (%xsd-string;)>
<!ELEMENT GPS_DEST_LAT_REF                  (%xsd-string;)>
<!ELEMENT GPS_DEST_LATITUDE                 (D, M, S?)>
<!ELEMENT GPS_DEST_LONG_REF                 (%xsd-string;)>
<!ELEMENT GPS_DEST_LONGITUDE                (D, M, S?)>
<!ELEMENT GPS_DEST_BEARING_REF              (%xsd-string;)>
<!ELEMENT GPS_DEST_BEARING                  (%tNonNegativeDouble;)>
<!ELEMENT GPS_DEST_DISTANCE_REF             (%xsd-string;)>
<!ELEMENT GPS_DEST_DISTANCE                 (%tNonNegativeDouble;)>

<!ELEMENT D                                 (%xsd-nonNegativeInteger;)>
<!ELEMENT M                                 (%xsd-nonNegativeInteger;)>
<!ELEMENT S                                 (%tNonNegativeDouble;)>


<!--
   - See section F.2.16  Direction Type
  -->
<!ENTITY % dig35-tDirection                 "(YAW?, PITCH?, ROLL?, COMMENT?)">
<!ELEMENT DIRECTION                         %dig35-tDirection;>
<!ATTLIST DIRECTION                         %att-lang-ts;>

<!ELEMENT YAW                               (%tDegree;)>
<!ELEMENT PITCH                             (%tHalfDegree;)>
<!ELEMENT ROLL                              (%tDegree;)>


<!--
   - See section F.2.17  Position Type
  -->
<!ENTITY % dig35-tPosition                  "((POINT | RECT | (RECT, REGION))?,
                                            COMMENT?)">
<!ELEMENT POSITION                          %dig35-tPosition;>
<!ATTLIST POSITION                          %att-lang-ts;>


<!--
   - See section F.2.17.1  Point Type
  -->
<!ELEMENT POINT                             (X, Y)>


<!--
   - See section F.2.17.2  Rectangle Type
  -->
<!ELEMENT RECT                              (X, Y, WIDTH, HEIGHT)>


<!--
   - See section F.2.17.3  Region Type
  -->
<!ELEMENT REGION                            (POINT, (POINT | SPLINE)*)>
<!ELEMENT SPLINE                            (X1, Y1, X2, Y2, X, Y)>

<!ELEMENT X                                 (%tNonNegativeDouble;)>
```

```
<!ELEMENT Y                                (%tNonNegativeDouble;)>
<!ELEMENT WIDTH                            (%tNonNegativeDouble;)>
<!ELEMENT HEIGHT                           (%tNonNegativeDouble;)>
<!ELEMENT X1                               (%tNonNegativeDouble;)>
<!ELEMENT Y1                               (%tNonNegativeDouble;)>
<!ELEMENT X2                               (%tNonNegativeDouble;)>
<!ELEMENT Y2                               (%tNonNegativeDouble;)>


<!--
   - See section F.2.18  Product Details Type
  -->
<!ENTITY % dig35-tProductDetails          "(MANUFACTURER?, MODEL?, SERIAL?,
                                            VERSION?)">

<!ELEMENT MANUFACTURER                     %dig35-tOrganization;>
<!ATTLIST MANUFACTURER                     %att-lang-ts-id;>
<!ELEMENT MODEL                            (%xsd-string;)>
<!ELEMENT SERIAL                           (%xsd-string;)>
<!ELEMENT VERSION                          (%xsd-string;)>


<!--
   - See section F.4.1  Comment
  -->
<!ELEMENT COMMENT                          (%xsd-string;)>
<!ATTLIST COMMENT                          %att-lang-ts;>



<!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - See section Annex A:  Basic Image Parameter Metadata
  -->
<!ELEMENT BASIC_IMAGE_PARAM                (BASIC_IMAGE_INFO?,
                                            PREF_PRESENTATION_PARAM?,
                                            COLOR_INFO?, COMPONENT_INFO?)>
<!ATTLIST BASIC_IMAGE_PARAM                 %att-lang-ts;>


<!--
   - See section A.3.1  Basic Image Information
  -->
<!ELEMENT BASIC_IMAGE_INFO                 (FILE_FORMAT?, IMAGE_ID?,
                                            IMAGE_SIZE?, COMPRESSION?)>
<!ATTLIST BASIC_IMAGE_INFO                  %att-lang-ts;>


<!--
   - See section A.3.1.1  File and Format
  -->
<!ELEMENT FILE_FORMAT                      (FILE_NAME?, FORMAT_TYPE?,
                                            MIME_TYPE?, VERSION?)>

<!ELEMENT FILE_NAME                         (%xsd-uriReference;)>
<!ELEMENT FORMAT_TYPE                       (%xsd-string;)>
<!ELEMENT MIME_TYPE                         (%xsd-string;)>


<!--
   - See section A.3.1.2  Image Identifier
  -->
<!ELEMENT IMAGE_ID                          (UID?, ID_TYPE?)>
<!ELEMENT UID                               (%xsd-string;)>
<!ELEMENT ID_TYPE                           (%xsd-uriReference;)>


<!--
   - See section A.3.1.3  Image Size
  -->
<!ELEMENT IMAGE_SIZE                        %size;>
```

```
<!--
   - See section A.3.1.4  Compression Method
  -->
<!ELEMENT COMPRESSION                        (%xsd-string;)>


<!--
   - See section A.3.2  Preferred Presentation Parameters
  -->
<!ELEMENT PREF_PRESENTATION_PARAM           %size;>


<!--
   - See section A.3.3  Color Information
  -->
<!ELEMENT COLOR_INFO                        (COLORSPACE+)>
<!ATTLIST COLOR_INFO                        %att-lang-ts;>


<!--
   - See section A.3.3.1  Colorspace
  -->
<!ELEMENT COLORSPACE                        (PROFILE_NAME?, PROFILE_REF?)>

<!ELEMENT PROFILE_NAME                      (%xsd-string;)>
<!ATTLIST PROFILE_NAME                      %att-lang;>

<!ELEMENT PROFILE_REF                       (%xsd-uriReference;)>


<!--
   - See section A.3.4  Component Information
  -->
<!ELEMENT COMPONENT_INFO                   (NUM_COMPONENT?, PREMULTIPLIED?,
                                            COMPONENTS?, COMP_SIZE*)>
<!ATTLIST COMPONENT_INFO                    %att-lang-ts;>

<!ELEMENT NUM_COMPONENT                     (%xsd-positiveInteger;)>
<!ELEMENT PREMULTIPLIED                     (%xsd-boolean;)>
<!ELEMENT COMPONENTS                        (%xsd-string;)>
<!ELEMENT COMP_SIZE                         (%xsd-positiveInteger;)>



<!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - See section Annex B:  Image Creation Metadata
  -->
<!ELEMENT IMAGE_CREATION                   (GENERAL_CREATION_INFO?,
                                            CAMERA_CAPTURE?,
                                            SCANNER_CAPTURE?,
                                            CAPTURED_ITEM?)>
<!ATTLIST IMAGE_CREATION                    %att-lang-ts;>


<!--
   - See section B.3.1  General Creation Information
  -->
<!ELEMENT GENERAL_CREATION_INFO            (CREATION_TIME?, IMAGE_SOURCE?,
                                            SCENE_TYPE?, IMAGE_CREATOR?,
                                            OPERATOR_ORG?, OPERATOR_ID?)>
<!ATTLIST GENERAL_CREATION_INFO             %att-lang-ts;>

<!ELEMENT CREATION_TIME                     (%xsd-timeInstant;)>

<!ELEMENT IMAGE_SOURCE                      (%xsd-string;)>
<!ATTLIST IMAGE_SOURCE                      %att-lang;>

<!ELEMENT SCENE_TYPE                        (%xsd-string;)>
```

```
<!ATTLIST SCENE_TYPE                          %att-lang;>

<!ELEMENT IMAGE_CREATOR                       %dig35-tPerson;>
<!ATTLIST IMAGE_CREATOR                       %att-lang-ts-id;>

<!ELEMENT OPERATOR_ORG                        %dig35-tOrganization;>
<!ATTLIST OPERATOR_ORG                        %att-lang-ts-id;>

<!ELEMENT OPERATOR_ID                         (%xsd-string;)>
<!ATTLIST OPERATOR_ID                         %att-lang;>


<!--
   - See section B.3.2  Camera Capture
  -->
<!ELEMENT CAMERA_CAPTURE                     (CAMERA_INFO?, SOFTWARE_INFO?,
                                              LENS_INFO?, DEVICE_CHARACTER?,
                                              CAMERA_SETTINGS?, ACCESSORY*)>
<!ATTLIST CAMERA_CAPTURE                      %att-lang-ts;>

<!ELEMENT CAMERA_INFO                         %dig35-tProductDetails;>
<!ATTLIST CAMERA_INFO                         %att-lang-ts;>

<!ELEMENT SOFTWARE_INFO                       %dig35-tProductDetails;>
<!ATTLIST SOFTWARE_INFO                       %att-lang-ts;>

<!ELEMENT LENS_INFO                           %dig35-tProductDetails;>
<!ATTLIST LENS_INFO                           %att-lang-ts;>


<!--
   - See section B.3.2.4  Device Characterization
  -->
<!ELEMENT DEVICE_CHARACTER                    (SENSOR_TECHNOLOGY?,
                                              FOCAL_PLANE_RES?,
                                              SPECTRAL_SENSITIVITY?,
                                              ISO_SATURATION?, ISO_NOISE?,
                                              SPATIAL_FREQ_RESPONSE?,
                                              CFA_PATTERN?, OECF?, MIN_F_NUMBER?)>
<!ATTLIST DEVICE_CHARACTER                    %att-lang-ts;>

<!ELEMENT SENSOR_TECHNOLOGY                   (%xsd-string;)>

<!ELEMENT FOCAL_PLANE_RES                     %size;>

<!ELEMENT SPECTRAL_SENSITIVITY                ANY>
<!ELEMENT ISO_SATURATION                      (%tNonNegativeDouble;)>
<!ELEMENT ISO_NOISE                           (%tNonNegativeDouble;)>

<!ELEMENT SPATIAL_FREQ_RESPONSE               (SPATIAL_FREQ_VAL+)>
<!ELEMENT SPATIAL_FREQ_VAL                    (SPATIAL_FREQ, HORIZ_SFR, VERT_SFR)>
<!ELEMENT SPATIAL_FREQ                        (%tNonNegativeDouble;)>
<!ELEMENT HORIZ_SFR                           (%tNonNegativeDouble;)>
<!ELEMENT VERT_SFR                            (%tNonNegativeDouble;)>

<!ELEMENT CFA_PATTERN                         (COLOR_ROW+)>
<!ELEMENT COLOR_ROW                           (COLOR+)>
<!ELEMENT COLOR                               (%xsd-string;)>

<!ELEMENT OECF                                (LOG_VAL+)>
<!ELEMENT LOG_VAL                             (LOG_EXPOSURE, OUTPUT_LEVEL+)>
<!ELEMENT LOG_EXPOSURE                        (%xsd-double;)>
<!ELEMENT OUTPUT_LEVEL                        (%tNonNegativeDouble;)>

<!ELEMENT MIN_F_NUMBER                        (%tNonNegativeDouble;)>


<!--
   - See section B.3.2.5  Camera Capture Settings
  -->
```

```
<!ELEMENT CAMERA_SETTINGS                  ((EXP_TIME | R_EXP_TIME)?,
                                            F_NUMBER?, EXP_PROGRAM?,
                                            BRIGHTNESS?, EXPOSURE_BIAS?,
                                            SUBJECT_DISTANCE?, METERING_MODE?,
                                            SCENE_ILLUMINANT?, COLOR_TEMP?,
                                            FOCAL_LENGTH?, FLASH?,
                                            FLASH_ENERGY?, FLASH_RETURN?,
                                            BACK_LIGHT?, SUBJECT_POSITION?,
                                            EXPOSURE_INDEX?, AUTO_FOCUS?,
                                            SPECIAL_EFFECT*, CAMERA_LOCATION?,
                                            ORIENTATION?, PAR?)>
<!ATTLIST CAMERA_SETTINGS                   %att-lang-ts;>

<!ELEMENT EXP_TIME                          (%tNonNegativeDouble;)>
<!ELEMENT R_EXP_TIME                        (%tRational;)>
<!ELEMENT F_NUMBER                          (%tNonNegativeDouble;)>
<!ELEMENT EXP_PROGRAM                       (%xsd-string;)>
<!ATTLIST EXP_PROGRAM                       %att-lang;>
<!ELEMENT BRIGHTNESS                        (%xsd-double;)>
<!ELEMENT EXPOSURE_BIAS                     (%xsd-double;)>
<!ELEMENT SUBJECT_DISTANCE                  (%tNonNegativeDouble;)>
<!ELEMENT METERING_MODE                     (%xsd-string;)>
<!ATTLIST METERING_MODE                     %att-lang;>
<!ELEMENT SCENE_ILLUMINANT                  (%xsd-string;)>
<!ATTLIST SCENE_ILLUMINANT                  %att-lang;>
<!ELEMENT COLOR_TEMP                        (%tNonNegativeDouble;)>
<!ELEMENT FOCAL_LENGTH                      (%tNonNegativeDouble;)>
<!ELEMENT FLASH                             (%xsd-string;)>
<!ELEMENT FLASH_ENERGY                      (%tNonNegativeDouble;)>
<!ELEMENT FLASH_RETURN                      (%xsd-boolean;)>
<!ELEMENT BACK_LIGHT                        (%xsd-string;)>
<!ELEMENT SUBJECT_POSITION                  %dig35-tPosition;>
<!ATTLIST SUBJECT_POSITION                  %att-lang-ts;>
<!ELEMENT EXPOSURE_INDEX                    (%xsd-double;)>
<!ELEMENT AUTO_FOCUS                        (%xsd-string;)>
<!ELEMENT SPECIAL_EFFECT                    (%xsd-string;)>
<!ELEMENT CAMERA_LOCATION                   %dig35-tLocation;>
<!ATTLIST CAMERA_LOCATION                   %att-lang-ts;>
<!ELEMENT ORIENTATION                       %dig35-tDirection;>
<!ATTLIST ORIENTATION                       %att-lang-ts;>
<!ELEMENT PAR                               (%tRational;)>

<!ELEMENT ACCESSORY                         %dig35-tProductDetails;>
<!ATTLIST ACCESSORY                         %att-lang-ts;>


<!--
   - See section B.3.3  Scanner Capture
  -->
<!ELEMENT SCANNER_CAPTURE                  (SCANNER_INFO?, SOFTWARE_INFO?,
                                            SCANNER_SETTINGS?)>
<!ATTLIST SCANNER_CAPTURE                   %att-lang-ts;>

<!ELEMENT SCANNER_INFO                      %dig35-tProductDetails;>
<!ATTLIST SCANNER_INFO                      %att-lang-ts;>


<!--
   - See section B.3.3.3  Scanner Capture Settings
  -->
<!ELEMENT SCANNER_SETTINGS                  (PIXEL_SIZE?, PHYSICAL_SCAN_RES?)>
<!ATTLIST SCANNER_SETTINGS                  %att-timestamp;>

<!ELEMENT PIXEL_SIZE                        (%tNonNegativeDouble;)>
<!ELEMENT PHYSICAL_SCAN_RES                 %size;>


<!--
   - See section B.3.4  Captured Item
  -->
```

```
<!ELEMENT CAPTURED_ITEM                    (REFLECTION_PRINT | FILM)>
<!ATTLIST CAPTURED_ITEM                    %att-lang-ts;>


<!--
   - See section B.3.4.1  Reflection Print
  -->
<!ELEMENT REFLECTION_PRINT                 (DOCUMENT_SIZE?, MEDIUM?, RP_TYPE?)>

<!ELEMENT DOCUMENT_SIZE                    %size;>
<!ELEMENT MEDIUM                           (%xsd-string;)>
<!ELEMENT RP_TYPE                          (%xsd-string;)>


<!--
   - See section B.3.4.2  Film
  -->
<!ELEMENT FILM                            (BRAND?, CATEGORY?, FILM_SIZE?,
                                           ROLL_ID?, FRAME_ID?, FILM_SPEED?)>
<!ATTLIST FILM                             %att-lang-ts;>

<!ELEMENT BRAND                            %dig35-tProductDetails;>
<!ATTLIST BRAND                            %att-lang-ts;>
<!ELEMENT CATEGORY                         (%xsd-string;)>
<!ELEMENT FILM_SIZE                        %size;>
<!ELEMENT ROLL_ID                          (%xsd-string;)>
<!ATTLIST ROLL_ID                          %att-lang;>
<!ELEMENT FRAME_ID                         (%xsd-positiveInteger;)>
<!ELEMENT FILM_SPEED                       (%xsd-positiveInteger;)>



<!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - See section Annex C:  Content Description Metadata
  -->
<!ELEMENT CONTENT_DESCRIPTION              (GROUP_CAPTION?, CAPTION?,
                                           CAPTURE_TIME?, LOCATION?,
                                           PERSON*, THING*, ORGANIZATION*,
                                           EVENT*, AUDIO*, PROPERTY*,
                                           DICTIONARY*, COMMENT?)>
<!ATTLIST CONTENT_DESCRIPTION              %att-lang-ts;>


<!ELEMENT GROUP_CAPTION                    (%xsd-string;)>
<!ATTLIST GROUP_CAPTION                    %att-lang;>

<!ELEMENT CAPTION                          (%xsd-string;)>
<!ATTLIST CAPTION                          %att-lang;>

<!ELEMENT CAPTURE_TIME                     (%dig35-tDateTime;)>
<!ATTLIST CAPTURE_TIME                     %att-lang-ts;>


<!--
   - See section C.3.5  Person Description
  -->
<!ELEMENT PERSON                          (%dig35-tPerson;, POSITION?,
                                           LOCATION?, PROPERTY*)>
<!ATTLIST PERSON                           %att-lang-ts-id;>


<!--
   - See section C.3.6  Thing Description
  -->
<!ELEMENT THING                            (NAME?, COMMENT?, POSITION?,
                                            LOCATION?, PROPERTY*, THING*)>
<!ATTLIST THING                            %att-lang-ts-id;>


<!--
```

```
   - See section C.3.7  Organization Description
  -->
<!ELEMENT ORGANIZATION                     (%dig35-tOrganization;, POSITION?,
                                            LOCATION?, PROPERTY*)>
<!ATTLIST ORGANIZATION                     %att-lang-ts-id;>


<!--
   - See section C.3.8  Event Description
  -->
<!ELEMENT EVENT                            (EVENT_TYPE, DESCRIPTION?,
                                            LOCATION?, EVENT_TIME?, DURATION?,
                                            COMMENT?, PARTICIPANT*,
                                            EVENT_RELATION*,
                                            (EVENT | EVENT_REF)*)>
<!ATTLIST EVENT                            %att-lang-ts-id;>

<!ELEMENT EVENT_TYPE                       (%xsd-string;)>
<!ATTLIST EVENT_TYPE                       %att-lang;>

<!ELEMENT DESCRIPTION                      (%xsd-string;)>
<!ATTLIST DESCRIPTION                      %att-lang;>

<!ELEMENT EVENT_TIME                       (%dig35-tDateTime;)>
<!ATTLIST EVENT_TIME                       %att-lang-ts;>

<!ELEMENT DURATION                         (%xsd-timeDuration;)>

<!ELEMENT PARTICIPANT                      (ROLE+, OBJECT_REF)>
<!ATTLIST PARTICIPANT                      %att-lang;>

<!ELEMENT ROLE                             (%xsd-string;)>
<!ATTLIST ROLE                             %att-lang;>

<!ELEMENT OBJECT_REF                       (%xsd-string;)>

<!ELEMENT EVENT_RELATION                   (RELATION*, EVENT_REF+)>

<!ELEMENT RELATION                         (%xsd-string;)>
<!ATTLIST RELATION                         %att-lang;>

<!ELEMENT EVENT_REF                        (%xsd-string;)>


<!--
   - See section C.3.9  Audio
  -->
<!ELEMENT AUDIO                            (AUDIO_STREAM?, AUDIO_FORMAT?,
                                           MIME_TYPE?, DESCRIPTION?, COMMENT?)>
<!ATTLIST AUDIO                            %att-lang-ts;>

<!ELEMENT AUDIO_STREAM                     (%xsd-uriReference;)>
<!ELEMENT AUDIO_FORMAT                     (%xsd-string;)>


<!--
   - See section C.3.10  Property
  -->
<!ELEMENT PROPERTY                         (NAME?, VALUE*, COMMENT?, PROPERTY*)>
<!ATTLIST PROPERTY                          %att-lang-ts;
                                            DICT_REF %xsd-string-c; #IMPLIED>

<!ELEMENT NAME                             (%xsd-string;)>
<!ATTLIST NAME                              %att-lang;>

<!ELEMENT VALUE                            (%xsd-string;)>
<!ATTLIST VALUE                             %att-lang;>


<!--
```

```
   - See section C.3.11  Dictionary Definition
  -->
<!ELEMENT DICTIONARY                       (DICT_NAME?, COMMENT?)>
<!ATTLIST DICTIONARY                        %att-lang-ts-id;>

<!ELEMENT DICT_NAME                         (%xsd-string;)>
<!ATTLIST DICT_NAME                         %att-lang;>



<!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - See section Annex D:  History Metadata
  -->
<!ELEMENT HISTORY                          (PROCESSING_SUMMARY?,
                                            IMAGE_PROCESSING_HINTS?, METADATA*)>
<!ATTLIST HISTORY                           %att-lang-ts;>


<!--
   - See section D.3.1  Processing Summary
  -->
<!ELEMENT PROCESSING_SUMMARY                (IMG_CREATED?, IMG_CROPPED?,
                                            IMG_TRANSFORMED?, IMG_GTC_ADJ?,
                                            IMG_STC_ADJ?, IMG_SPATIAL_ADJ?,
                                            IMG_EXT_EDITED?, IMG_RETOUCHED?,
                                            IMG_COMPOSITED?, IMG_METADATA?)>
<!ATTLIST PROCESSING_SUMMARY                %att-timestamp;>


<!--
   - See section D.3.2  Image Processing Hints
  -->
<!ELEMENT IMAGE_PROCESSING_HINTS           (IMG_CREATED | IMG_CROPPED |
                                            IMG_TRANSFORMED | IMG_GTC_ADJ |
                                            IMG_STC_ADJ | IMG_SPATIAL_ADJ |
                                            IMG_EXT_EDITED | IMG_RETOUCHED |
                                            IMG_COMPOSITED | IMG_METADATA)*>
<!ATTLIST IMAGE_PROCESSING_HINTS            %att-lang-ts;>

<!ELEMENT IMG_CREATED                       (%xsd-string;)>
<!ELEMENT IMG_CROPPED                       (%xsd-string;)>
<!ELEMENT IMG_TRANSFORMED                   (%xsd-string;)>
<!ELEMENT IMG_GTC_ADJ                       (%xsd-string;)>
<!ELEMENT IMG_STC_ADJ                       (%xsd-string;)>
<!ELEMENT IMG_SPATIAL_ADJ                   (%xsd-string;)>
<!ELEMENT IMG_EXT_EDITED                    (%xsd-string;)>
<!ELEMENT IMG_RETOUCHED                     (%xsd-string;)>
<!ELEMENT IMG_COMPOSITED                    (%xsd-string;)>
<!ELEMENT IMG_METADATA                      (%xsd-string;)>



<!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - See section Annex E:  Intellectual Property Rights Metadata
  -->
<!ELEMENT IPR                              (IPR_NAMES?, IPR_DESCRIPTION?,
                                            IPR_DATES?, IPR_EXPLOITATION?,
                                            IPR_IDENTIFICATION?,
                                            IPR_CONTACT_POINT?, IPR_HISTORY?)>
<!ATTLIST IPR                               %att-lang-ts;>


<!--
   - See section E.3.1  Names
  -->
<!ELEMENT IPR_NAMES                        (IPR_PERSON?, IPR_ORG?,
                                            IPR_NAME_REF?)+>
<!ATTLIST IPR_NAMES                         %att-lang-ts;>

<!ELEMENT IPR_PERSON                        %dig35-tPerson;>
```

```
<!ATTLIST IPR_PERSON                        DESCRIPTION %xsd-string-c; #IMPLIED
                                            %att-lang-ts-id;>

<!ELEMENT IPR_ORG                           %dig35-tOrganization;>
<!ATTLIST IPR_ORG                           DESCRIPTION %xsd-string-c; #IMPLIED
                                            %att-lang-ts-id;>

<!ELEMENT IPR_NAME_REF                      (%xsd-string;)>
<!ATTLIST IPR_NAME_REF                      DESCRIPTION %xsd-string-c; #IMPLIED>


<!--
   - See section E.3.2  Description
  -->
<!ELEMENT IPR_DESCRIPTION                  (IPR_TITLE?, IPR_LEGEND?,
                                            IPR_CAPTION?, COPYRIGHT?)>

<!ELEMENT IPR_TITLE                         (%xsd-string;)>
<!ATTLIST IPR_TITLE                         %att-lang-ts;>

<!ELEMENT IPR_LEGEND                        (%xsd-string;)>
<!ATTLIST IPR_LEGEND                        %att-lang-ts;>

<!ELEMENT IPR_CAPTION                       (%xsd-string;)>
<!ATTLIST IPR_CAPTION                       %att-lang-ts;>

<!ELEMENT COPYRIGHT                         (%xsd-string;)>
<!ATTLIST COPYRIGHT                         %att-lang-ts;>


<!--
   - See section E.3.3  Dates
  -->
<!ELEMENT IPR_DATES                         (IPR_DATE+)>
<!ATTLIST IPR_DATES                         %att-lang-ts;>

<!ELEMENT IPR_DATE                          (%dig35-tDateTime;)>
<!ATTLIST IPR_DATE                          DESCRIPTION %xsd-string-c; #IMPLIED
                                           %att-lang-ts;>


<!--
   - See section E.3.4  Exploitation
  -->
<!ELEMENT IPR_EXPLOITATION                 (IPR_PROTECTION?,
                                            IPR_USE_RESTRICTION?,
                                            IPR_OBLIGATION?,
                                            IPR_MGMT_SYS?)>
<!ATTLIST IPR_EXPLOITATION                  %att-lang-ts;>

<!ELEMENT IPR_PROTECTION                    (%xsd-nonNegativeInteger;)>

<!ELEMENT IPR_USE_RESTRICTION               (%xsd-string;)>
<!ATTLIST IPR_USE_RESTRICTION               %att-lang;>

<!ELEMENT IPR_OBLIGATION                    (%xsd-string;)>
<!ATTLIST IPR_OBLIGATION                    %att-lang;>


<!--
   - See section E.3.4.4  IPR Management System
  -->
<!ELEMENT IPR_MGMT_SYS                     (IPR_MGMT_TYPE?,
                                            IPR_MGMT_SYS_ID?,
                                            IPR_MGMT_SYS_LOCATION?)>
<!ATTLIST IPR_MGMT_SYS                      %att-lang-ts;>

<!ELEMENT IPR_MGMT_TYPE                     (%xsd-string;)>
<!ELEMENT IPR_MGMT_SYS_ID                   (%xsd-string;)>
```

```
<!ELEMENT IPR_MGMT_SYS_LOCATION             (%xsd-uriReference;)>


<!--
   - See section E.3.5  Identification
  -->
<!ELEMENT IPR_IDENTIFICATION               (IPR_IDENTIFIER?,
                                            LICENCE_PLATE?)>
<!ATTLIST IPR_IDENTIFICATION                %att-lang-ts;>


<!--
   - See section E.3.5.1  Generic IPR Identifier
  -->
<!ELEMENT IPR_IDENTIFIER                    (IPR_ID_MODE?, IPR_ID?)>

<!ELEMENT IPR_ID_MODE                       (%xsd-string;)>
<!ATTLIST IPR_ID_MODE                       %att-lang;>
<!ELEMENT IPR_ID                            (%xsd-string;)>
<!ATTLIST IPR_ID                            %att-lang;>


<!--
   - See section E.3.5.2  Licence Plate
  -->
<!ELEMENT LICENCE_PLATE                    (LP_COUNTRY?,
                                            LP_REG_AUT?,
                                            LP_REG_NUM?,
                                            LP_DELIVERY_DATE?)>

<!ELEMENT LP_COUNTRY                        (%xsd-string;)>
<!ELEMENT LP_REG_AUT                        (%xsd-string;)>
<!ELEMENT LP_REG_NUM                        (%xsd-string;)>
<!ELEMENT LP_DELIVERY_DATE                  (%xsd-timeInstant;)>


<!--
   - See section E.3.6  Contact Point
  -->
<!ELEMENT IPR_CONTACT_POINT               (IPR_PERSON | IPR_ORG
                                          | IPR_NAME_REF)>
<!ATTLIST IPR_CONTACT_POINT                %att-lang-ts;>


<!--
   - See section E.3.7  IPR History
  -->
<!ELEMENT IPR_HISTORY                      (IPR+)>
<!ATTLIST IPR_HISTORY                      %att-lang-ts;>
```

APPENDICIES

Appendix I: Examples

I.1 Metadata Examples

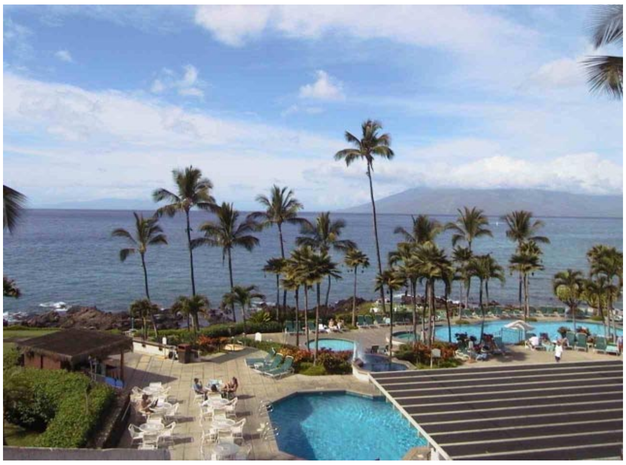


Figure I-1: Example image

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE METADATA SYSTEM "-//DIG//DTD DIG35 1.0//EN" "dig35.dtd">

<METADATA>
    <CREATION>
        <GENERAL_CREATION>
            <CAPTURE_TIME>
                <EXACT>1999-12-9T12:30:31</EXACT>
            </CAPTURE_TIME>
            <IMAGE_CREATOR>
                <PERSON_NAME>
                    <NC TYPE="Given">Katsuki</NC>
                    <NC TYPE="Family">Ishii</NC>
                </PERSON_NAME>
                <NICKNAME>Kats</NICKNAME>
            </IMAGE_CREATOR>
        </GENERAL_CREATION>
        <CAMERA_CAPTURE>
            <CAMERA_INFO>
                <MANUFACTURER>
                    <ORG_NAME>Acme</ORG_NAME>
                </MANUFACTURER>
                <MODEL>Model 1000</MODEL>
            </CAMERA_INFO>
            <CAMERA_SETTINGS>
                <EXP_TIME>1/60</EXP_TIME>
                <F_NUMBER>5.6</F_NUMBER>
                <EXPOSURE_BIAS>0</EXPOSURE_BIAS>
                <SUBJECT_DISTANCE>10</SUBJECT_DISTANCE>
                <FLASH>FALSE</FLASH>
            </CAMERA_SETTINGS>
        </CAMERA_CAPTURE>
    </CREATION>
    <CONTENT>
        <CAPTION>View from restaurant</CAPTION>
        <COMMENT>A very nice day.</COMMENT>
        <LOCATION>
            <ADDRESS>
                <ADDR TYPE="City">Maui</ADDR>
                <COUNTRY>US</COUNTRY>
            </ADDRESS>
        </LOCATION>
        <EVENT>
            <EVENT_TYPE>Meeting</EVENT_TYPE>
            <DESCRIPTION>Committee Meeting</DESCRIPTION>
        </EVENT>
    </CONTENT>
</METADATA>
```

I.2 Internal Association Examples

This section discusses several methods to associate XML encoded metadata with various image file formats — most of which are currently implemented in digital cameras or supported by scanners, CD providers, online image processors, etc. This broad implementation at the input stage of imaging equipment allows developers to support the DIG35 Metadata and immediately take advantage of some metadata information. We further discuss the benefits and drawbacks of each method.

Many image file formats are commonly used in today's existing applications. The following table shows of the most common file formats that have association examples discussed in this section:

Table I-1: File Format and Association Method Relation

File format (variant)	Association method
Exif/TIFF (Uncompressed)	Using existing tags
	Using a Private Tag
Exif/JPEG (Compressed)	Using existing tags
	Using a Private Tag
	Using the APPn marker
TIFF	Using a Private Tag
Flashpix	Using existing property sets
	Using a new property set
JPEG2000[i]	Using the XML Box
	Defining DIG35 Boxes
Format independent	Using the end of file

Several alternatives for associating the internal metadata to DIG35 metadata may exist for each file format and the most useful method is application dependent. However, if those associated metadata are to be properly exchanged, the sending and receiving application must both comply with a common implementation method.

[i] This is based on the Final draft international standard dated 2000/8/18

I.2.1 File Format Dependent Association

I.2.1.1 Using Existing Exif/TIFF Tags

XML implementation requires multi-byte capability and extensible storage. Within Exif/TIFF there are only two tags that permit multi-byte and arbitrary size data: "MakerNote" and "UserComment." The MakerNote tag is reserved for manufacturers information. The DIG35 application therefore should store DIG35 metadata in the "UserComment" tag as defined in the Exif specification.

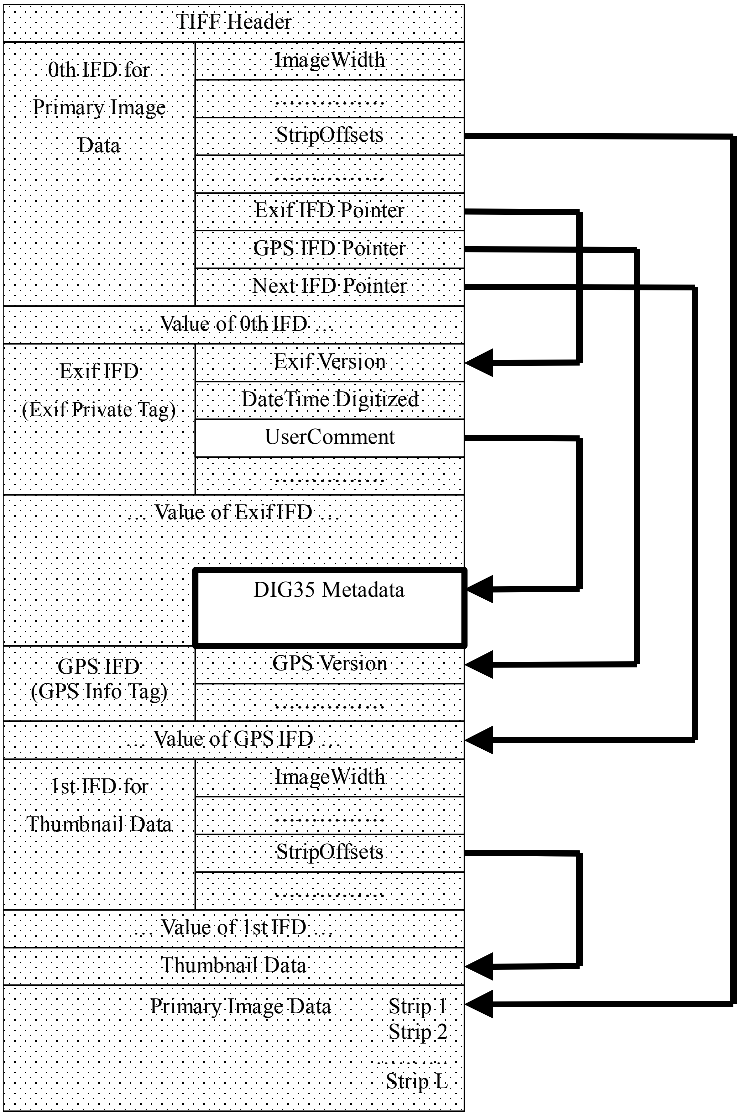


Figure I-2: Uncompressed Exif/TIFF file format with DIG35 metadata

Table I-2: Discussion items on Exif/TIFF using UserComment

Pros	- No need to define a new Tag
Cons	- If other applications use the "UserComment" tag for a different purpose, DIG35 metadata may not use this tag or there is a risk of being overwritten. - If user increase "UserComment" size, many offset tags need to recalculate each address.

I.2.1.2 Using Exif/TIFF Private Tag

DIG35 Metadata may be recorded using a "Private Tag" in the 0th IFD.

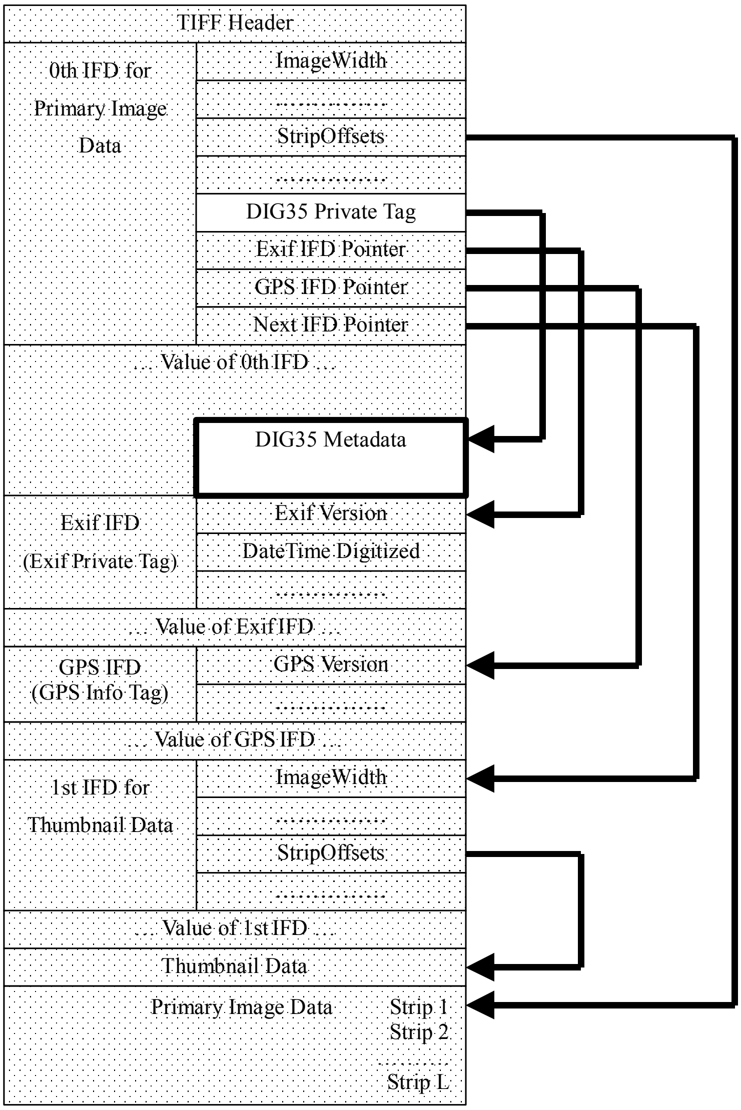


Figure I-3: Uncompressed Exif file format with DIG35 metadata

Table I-3: Discussion items on Exif/TIFF using Private Tag

Pros	- Only DIG35 Metadata uses the new private tag.
Cons	- Need to add a NEW private Tag. - If metadata are added increasing the size, many offset tags need to recalculate each address.

I.2.1.3 Using Existing Exif/JPEG Tag

The Exif/JPEG format has a similar situation to the Exif/TIFF format; therefore DIG35 metadata may be recorded in the "UserComment" tag defined in Exif IFD.

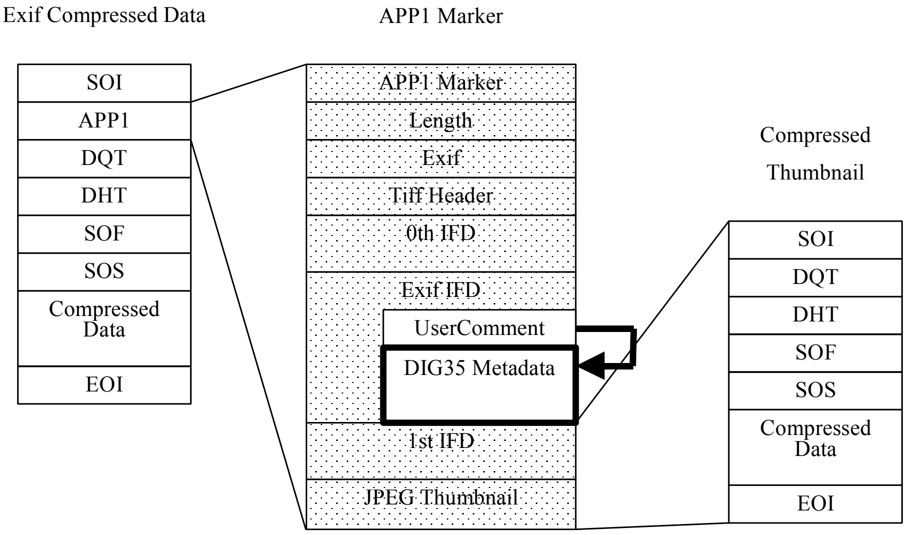


Figure I-4: Compressed Exif/JPEG file format with DIG35 metadata

Table I-4: Discussion items on Exif/JPEG using UserComment

Pros	- No need to define a new Tag
Cons	- If other applications use the "UserComment" tag for a different purpose, DIG35 metadata may not use this tag and has the risk of being overwritten. - If user increase "UserComment" size, many offset tags need to recalculate each address.

I.2.1.4 Using Exif/JPEG Private Tag

DIG35 Metadata may be recorded using a "Private tag" in the 0th IFD.

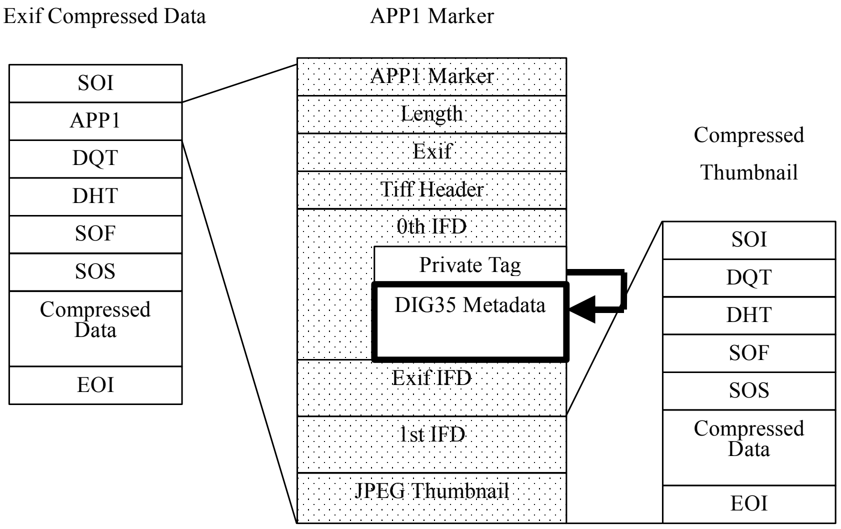


Figure I-5: Compressed Exif data files with DIG35 metadata

Table I-5: Discussion items on Exif/JPEG using Private Tag

Pros	- Only DIG35 Metadata uses the new private tag.
Cons	- Need to add a NEW private Tag. - If metadata are added increasing the size, many offset tags need to recalculate each address.

I.2.1.5 Using Exif/JPEG APPn marker

DIG35 Metadata may be recorded under a different APP marker. This example uses the APP4 marker.

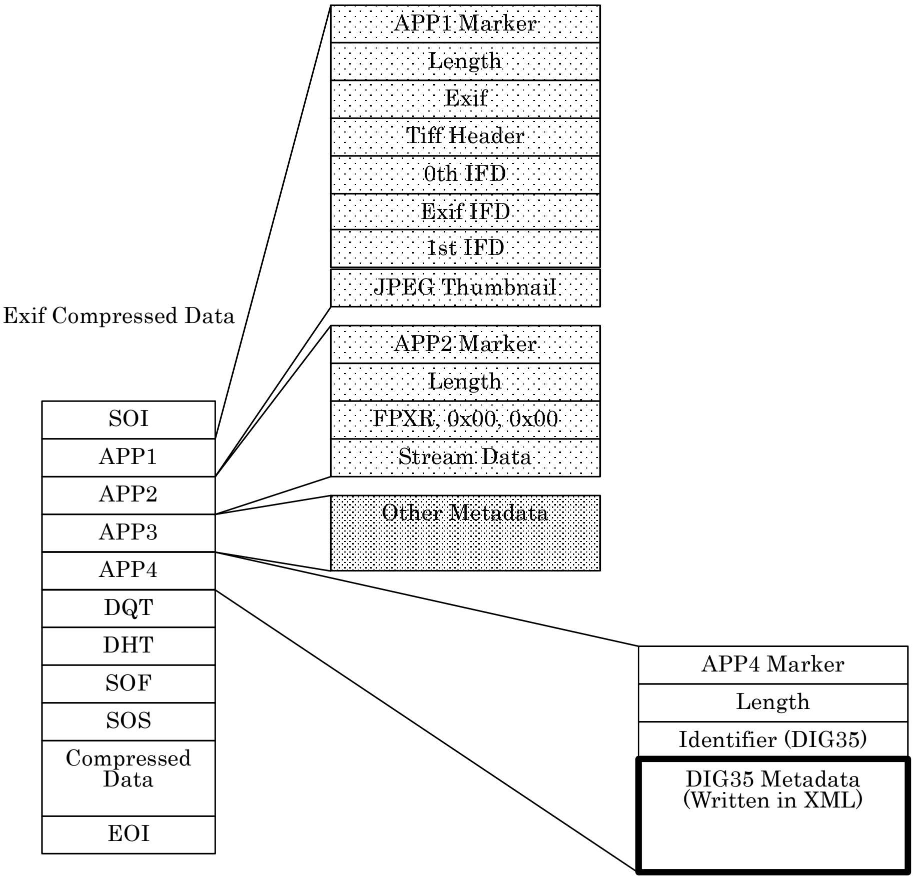


Figure I-6: Compressed Exif data files with DIG35 metadata

Table I-6: Discussion items on Exif/JPEG using APP marker

PROS	- Only DIG35 Metadata uses the defined APP Marker.
CONS	- Need to communicate the use of the APP Marker. - APP Marker maybe used or used differently in other applications.

I.2.1.6 Using TIFF Private Tag

DIG35 Metadata may be recorded using a “Private Tag” in a TIFF file.

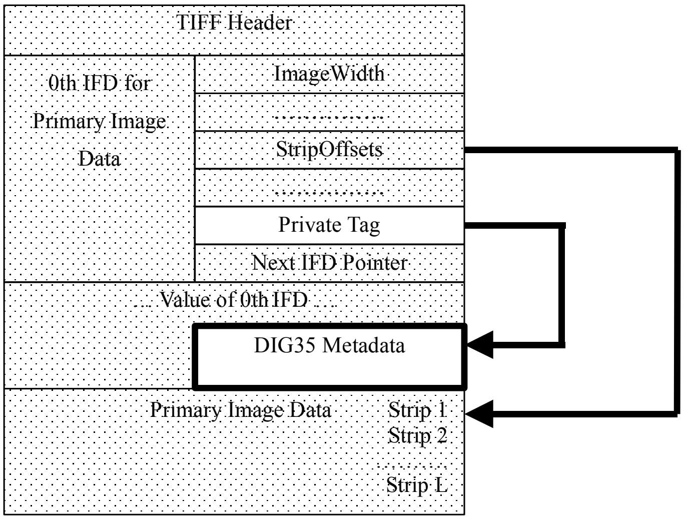


Figure I-7: TIFF file format with DIG35 metadata

Table I-7: Discussion items on TIFF using Private Tag

Pros	- A new private tag is allocated specifically used for DIG35 Metadata.
Cons	- Need to register a NEW private Tag ID.

I.2.1.7 Using Existing Flashpix Property Set

To record DIG35 Metadata in the Flashpix file format, it may be stored in the "Content description note" of "Content Description" property.

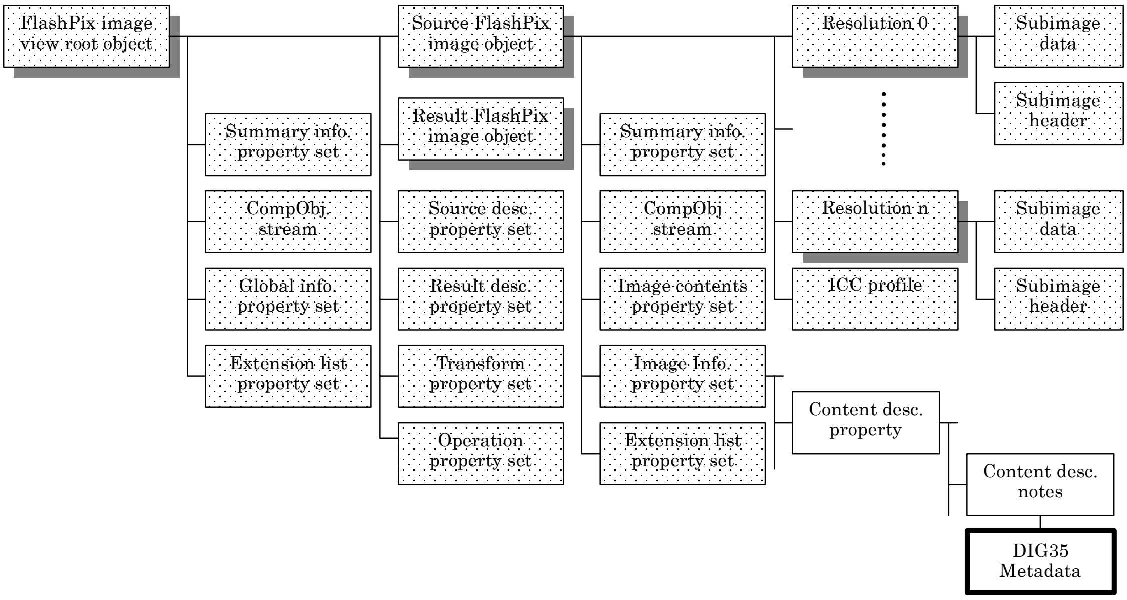

FlashPix image view root object
Summary info. property set
CompObj. stream
Global info. property set
Extension list property set
Source FlashPix image object
Result FlashPix image object
Source desc. property set
Result desc. property set
Transform property set
Operation property set
Summary info. property set
CompObj stream
Image contents property set
Image Info. property set
Extension list property set
Resolution 0
Resolution n
ICC profile
Subimage data
Subimage header
Subimage data
Subimage header
Content desc. property
Content desc. notes
DIG35 Metadata

Figure I-8: Flashpix file format with DIG35 metadata

Table I-8: Discussion items on Flashpix using Content Description Note

Pros	- No need to add a NEW property set. - Exif's "UserComment" can be converted to Flashpix's "Description Note" property.
Cons	- Most applications supporting Flashpix file format do not read/write the property set.

I.2.1.8 Defining a New Flashpix Property

DIG35 Metadata is recorded by defining a NEW property set. "DIG35 property set" for example, in the "Extension list".

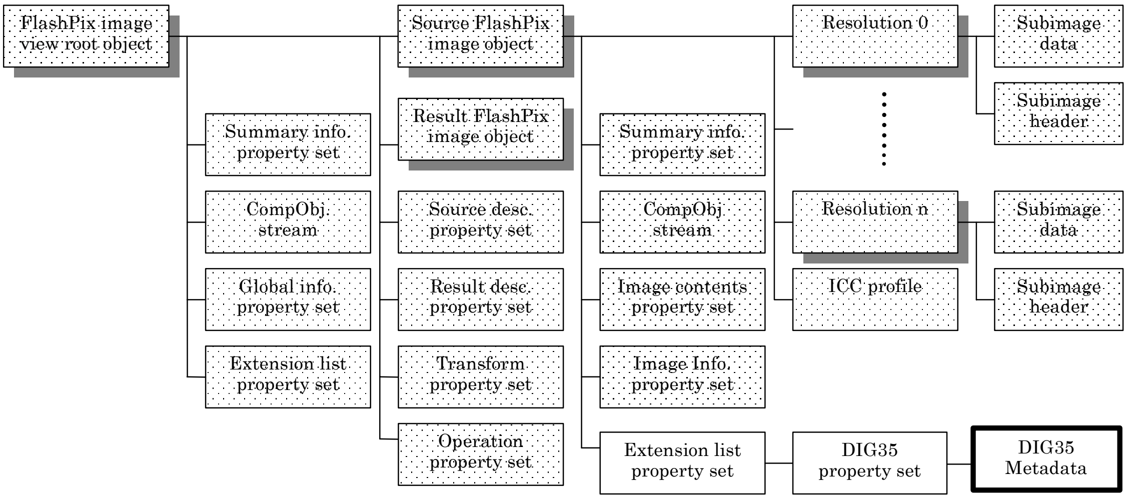

FlashPix image view root object
Summary info. property set
CompObj. stream
Global info. property set
Extension list property set
Source FlashPix image object
Result FlashPix image object
Source desc. property set
Result desc. property set
Transform property set
Operation property set
Summary info. property set
CompObj stream
Image contents property set
Image Info. property set
Extension list property set
Resolution 0
Resolution n
ICC profile
Subimage data
Subimage header
Subimage data
Subimage header
DIG35 property set
DIG35 Metadata

Figure I-9: Flashpix files format with DIG35 metadata

Table I-9: Discussion items on Flashpix defining a new property set

Pros	- Only DIG35 Metadata uses the new property set.
Cons	- Need to EXTEND the property set. - Most Flashpix file format supporting applications do not read/write the property set.

I.2.1.9 Using XML Box in JPEG2000 File Format

To record DIG35 Metadata in a JP2 file, it may be record in the “XML box”.

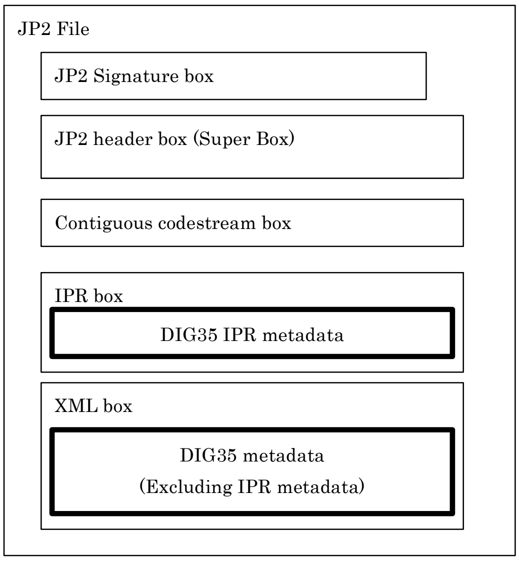


Figure I-10: JP2 file format with DIG35 metadata in XML box

Table I-10: Discussion items on using JP2 XML box

Pros	- JPEG2000 allows XML data (DIG35 Metadata) to be stored.
Cons	- May need to define XML Box / UUID boxes.

I.2.1.10 Define New Boxes for DIG35 Metadata

DIG or future revisions of DIG35 may define box types for each block in a JP2 file.

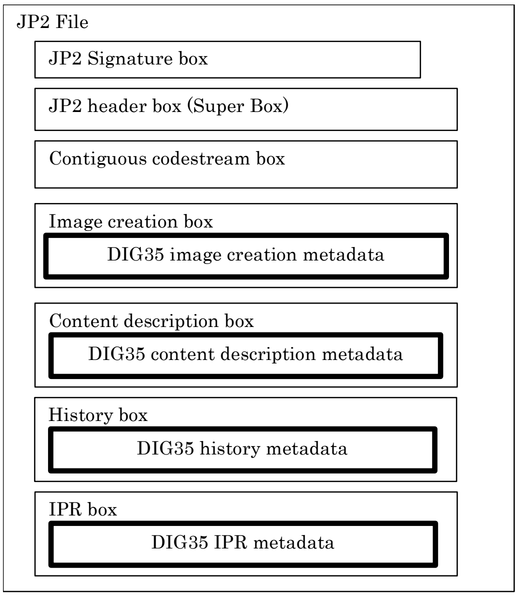


Figure I-11: JP2 file format with DIG35 metadata as separate boxes

Table I-11: Discussion items on defining separate DIG35 boxes for JP2

Pros	- DIG35 Metadata is identified by JPEG2000 box type
Cons	- Need to register box types to avoid box type collisions.

I.2.2 File Format Independent Association

I.2.2.1 File Format Independent

This is an example of a file format independent implementation. The metadata block consists of a header, the actual XML metadata and footer. The footer is used to identify images that are associated with metadata and those that are not. The identification method consists of reading the fixed length footer, then scan the header where you can calculate from the footer and verify the check-sum, and finally verify the metadata body. By this process, the reader is able to verify if the metadata is correctly set and properly detach the metadata from the image file.

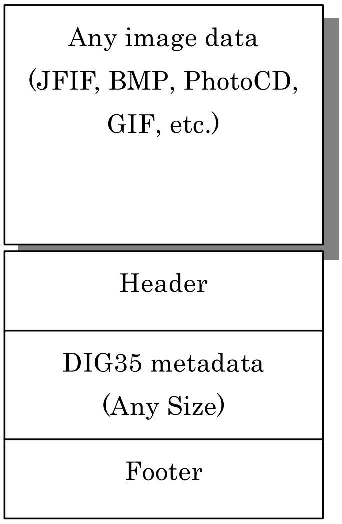


Figure I-12: A file format independent association method

Table I-12: Discussion items on File format independent association method

Pros	- Enable file format independent association - Can edit metadata without dealing with an image data. - Simple structure
Cons	- Becomes a different file as a whole (thus is not a legal internal association method; See Appendix II for detail discussions)

Appendix II: Metadata Association

II.1 Introduction

This section is intended to illustrate many of the challenges attributed to associating metadata with images and potential problems encountered relating to retention of image metadata throughout the lifecycle of that image.

DIG35 has identified three types of metadata association models: internal, external and hybrid (a combination of internal and external). All three types of association models are commonly used in the industry today.

(1) **Internal Model**: When using the internal model, metadata is stored with the image file itself. A good example is the storing of metadata in image file 'headers' or 'tags'. Image metadata may also be attached to a file even if a pre-existing mapping of the image file did not exist.

(2) **External Model**: The external model uses external references to associate metadata to an image file. Often times the image file name is used to link the external metadata and the image itself. Examples might be when an image is stored on a server while the metadata is stored locally.

(3) **Hybrid Model**: Using both internal and external metadata for association happens frequently. For example, some metadata (file headers/tags) are stored directly in the image file while additional descriptive metadata added later in the workflow is stored externally in a database.

As there are several challenges for each type of association models, each will be discussed in more detail below with examples of typical applications, challenges, and technical issues.

II.2 Internal Model

Using the internal model, image metadata is stored in the same file as the image data itself. Images may be tagged or appended with metadata; in either case metadata is stored within the image file. The internal model is a robust solution for applications in which there is a known image workflow in a controlled environment. It is not ideal for uncontrolled environments in which different file formats, devices and applications are used

II.2.1 Typical Applications

As discussed above, the 'internal model' is ideal for controlled environments where there may be only one file format, using a limited known set of devices and applications that are proven to be interoperable and to retain the internal metadata throughout the image workflow. This might be a business deployment of digital imaging hardware and applications, in which a single file format is allowed with a structured metadata template, and regulated applications. This type of environment ensures that all metadata is retained throughout the image workflow.

The internal model is less than ideal for a consumer-focused application in which a wide variety of capture devices, image editing application and services are used. In order to support all solutions used throughout the workflow, each device, application, or service must be aware of the metadata and allow for the retention and versioning of this metadata.

II.2.2 Challenges and Technical Issues

II.2.2.1 File Format Dependent

File formats may not support an internal metadata structure or metadata structures may differ from file format to file format. One way to combat this problem is to convert to a file format that supports an internal metadata structure (e.g. convert from JPEG to an Exif file for additional internal storage of metadata).

II.2.2.2 Metadata Aware Applications Required

Devices, applications, and services that work with a file that has internal metadata must be aware of this metadata and be able to retain, edit and version the metadata. Most applications in the industry today are *metadata unaware*, destroying, overwriting or ignoring internal descriptive information when a file is saved. Further, many software applications do not correctly retain non-proprietary metadata often overwriting this metadata with their own proprietary application tags. Replacing or revising such applications may be difficult.

II.2.2.3 Metadata Versioning

Defining how the previously stored metadata is retained, edited, or passed along in a given image workflow is a challenge. Furthermore, continuously adding metadata to the file results in a file size increase that will result in a file too large to use. However, maintaining previous metadata is important to many users and usages.

II.2.2.4 Metadata Sharing and Collections

Complexities also occur when a single set of metadata must be assigned to multiple images or groups of images. To support internal metadata, each image must be tagged individually, which is an inefficient process for a single set of metadata that can be associated with a group of images.

II.3 External Model

The external model provides a way to use external information along with a link to an image file in order to associate metadata with a digital image.

II.3.1 Typical Applications

Several common ways that the ‘external model’ is used today include using a database structure to identify and associate metadata with the image file or using an external file (such as a text file, HTML file, XML file). It is the DIG35’s goal to provide a common XML structure that can be used as a standard for external and internal model associations.

External models are ideal for using image files for one single application. For example, an online service that uses a database structure in order to store information about its users’ images is an example of an external model.

External models also fit well with images that are on read-only media such as CD-ROMs or DVDs. In these types of environments, images may stay on the read-only media, but the associated metadata must be assigned and managed else where in read-write form.

External models are less than ideal for users and workflow where there are many applications as the associated metadata files may become lost or broken when trying to tie a single image to multiple applications.

II.3.2 Challenges and Technical Issues

II.3.2.1 Broken Links

Using the external model, losing the link from an image file to a metadata file or from a database to an image file is a commonly known problem. This may occur if an image file is renamed and the corresponding metadata file or link is not updated to reflect this change. This is most likely to occur during the transmission of a digital image file (upload, download).

II.3.2.2 Different Association Types

As it is a challenge with the internal model to deal with the multiple file formats and their associated internal metadata structures, it is a challenge with the external model to deal with multiple association types (such as database, text file, etc.)

II.3.2.3 Metadata Versioning

Defining how previously entered metadata is retained, edited, or passed along in a given image workflow is a challenge in the external model also. Maintaining link integrity is a key challenge when working with image metadata using the external model.

II.3.2.4 Different Types of Metadata

The external model poses an additional challenge when the metadata that is associated is not textual metadata. For example, if the metadata is an audio annotation then the association must either be done via a file naming convention or using another external file such as a sound or text file.

II.4 Hybrid Model

The hybrid model uses a combination of both the internal and the external mechanisms for metadata association.

II.4.1 Typical Applications

Typical applications include a workflow in which image submissions are coming from many disparate sources, some of which are currently using the internal model and others that are using the external model.

II.4.2 Challenges and Technical Issues

II.4.2.1 Applications Must Support Both Types

Using both types of structures provides its own set of challenges. Applications must now support two different types of associations, requiring complex checking and validation and some intelligence about which takes precedence when both types exist.

II.4.2.2 Metadata Versioning

As mentioned in both the internal and external models, metadata versioning is key issue.

II.5 File Format Dependence

DIG35 and the DIG35 Recommended XML Implementation represent a file-format independent solution. The following tables illustrate the pros and cons of defining a standard that is either file format dependent or independent with relation to the models discussed above.

II.5.1 Internal model

	DEPENDENT	INDEPENDENT
PROS	- Legal image file format	- Single implementation applies to all formats - Append capability - Need not worry about different formats having different metadata fields defined
CONS	- Different file format may have different metadata schema. May not be able to pass one metadata set to another format. - Single API may be defined to attach DIG35 metadata, though need separate implementations for each file format to embed metadata	- Undefined format - May crash application - Image editing applications and services may need to be greatly redesigned.

II.5.2 External Model

	DEPENDENT	INDEPENDENT
PROS		- Support for all file formats regardless of internal structure.
CONS	- Limited file format support.	- Applications need to track file type or file extension

Appendix III: Conformance

Different sets of conformance criteria exist for supporting the metadata defined in this specification:

- Conforming metadata definitions
- Conforming metadata implementations

III.1 Conforming Metadata Definitions

An image metadata definition is a *DIG35 conforming metadata definition* if it adheres to the specification described in this document and also has the capability of converting the metadata to the recommended implementation format described in this document. Note that the metadata definitions shall *supersede* the recommended XML implementation in all cases.

Those who wish to conform at this level may choose a different implementation method other than XML.

III.2 Conforming Metadata Implementation

DIG35 metadata document is a *Conforming Metadata Implementation* if it adheres to the reference implementation and DTD specification described in this document and:

- is a well-formed XML document
- if all non-DIG35 namespace elements removed from the document, it is a valid XML document

III.2.1 Conforming DIG35 Stand-alone Files

It is a *Conforming DIG35 Stand-alone File* if:

- It is a conforming metadata implementation
- The root element is <METADATA>

III.2.2 Conforming DIG35 Included Files

A DIG35 metadata document that is included within another XML document is a *Conforming DIG35 Included File* if the DIG35 metadata document is properly extracted from the other XML document, conforms to the DIG35 DTD.

Appendix IV: Other Encoding Methods

While DIG35 chose XML as a recommended implementation method especially for image metadata interchange, there are other means to exchange, store or stream image metadata within various systems. Several alternatives include:

- Tag based binary encoding (e.g. TIFF)
- Simple KEY-VALUE pair plain text
- Relational/Object database systems
- Other proprietary formats

Appendix V: Future Plans

The following list consists of a list of technical issues not addressed by version 1.0 of this specification. A future version will consider the following.

V-1. For Content Metadata, define a unifying framework for both defined objects and user extensible objects in an image. The objects may be concrete or abstract.

V-2. Ability to compose a new object from other objects. (E.g. a car comprises an engine, wheels, tire, etc.)

V-3. Identifying semantic relationships between objects in an image. Examples of semantic relationships include husband, wife, teammate, co-worker, etc.

V-4. Ability to restrict or validate the participants in an event within an XML instance document.

V-5. Audio as metadata, audio as part of image.

V-6. Association scenarios when storage is non-writable.

V-7. Compression of individual blocks or the entire metadata.

V-8. Authentication of metadata.

V-9. Protection of metadata.

Appendix VI: Discussion Items

VI.1 Importance/Relevance

DIG35 have discussed the introduction of an importance attribute for elements. We have had the following suggestions for the name of the field:

"Importance"

"Relevance"

This attribute would specify which of the metadata is important (or relevant to the image), and which is not. For example, an image with a tree in the background, and a house in the foreground, may contain the following keywords with associated importance/relevance:

House (importance = 0.9)

Tree (importance = 0.1)

If a picture of a house used by a real estate agent, then the address field would be marked as important. Whereas if an image of a house was taken as the background of "Craig standing next to his new car", then the address might still be listed, but would be listed as unimportant.

There are a number of ideas under discussion for the value stored in the importance field:

1. Use a range of values. If a value of zero is used for unimportant, and 1 for important, then what does 0.5 mean? Also - how can we specify a system where different people classifying an importance of 0.5 in multiple image files that the meaning is equivalent?
2. Use a rating within a file where the most important item was listed as a 1, and the next most important - 2, with the least import item being a large number. However, this system would still not allow comparing the importance of items in separate files. This system also falls down in that the user cannot specify that all of the content description is unimportant.

Some research will be done to determine how other metadata systems specify importance.

VI.2 Non-validated metadata

It is not possible for the creators of any file format to insist that everyone who opens a file understand all the metadata included in the file. In fact, it is often true that the creators of file formats go to great lengths to insure that different levels of readers exist or that file writers can put private data in the file that others may not understand. Often, the only requirement is that readers pass over metadata they don't understand.

This, while necessary, has led to significant problems. In some cases, metadata that is not understood will be dropped when a file is written back out. In other cases when the metadata is not dropped it will be inconsistent.

In an effort to provide a third alternative, metadata that is not understood could be marked as "non-validated." This could be done by marking each piece of metadata or moving the metadata to a non-validated tree. In either case, subsequent readers would be able to scan the non-validated data for information that might be useful, but it would be able to tell what data the latest reader couldn't update.

A simple example would be a clip operation (although this problem could be avoided by other means). If an image editing program couldn't read tags left by earlier editing programs, it could mark the earlier programs information that it couldn't understand as invalid. These non-validated tags could include content information. It could then clip the image and write it out. If the earlier editing program was to re-open the image it could recover the content information, but the user would have to somehow re-validate it — but they would not have to regenerate or re-enter the data.

Appendix VII: Glossary

Basic Image Parameters: Technical information on the image data. For example, the colors depicted, or the number of pixels.

Capture Device: The equipment that generated (or captured) the image.

Content Description: Information on the items depicted in the image.

DIG: Digital Imaging Group

DTD: Document Type Definition

ECMS: Electronic Copyright Management System

Exif: Exchangeable Image File Format

File Format: The layout of a computer file (whether an image file or otherwise) is the file format. Files may be textual (human readable), or binary. The file format usually splits the file into segments of known content. File formats allow such functionality as “extensibility”, “random access”, and “interoperability.”

Quite often a file format will include a signature at the top of the file that specifies the format of the file. Examples include: JFIF, TIFF, XML

GPS: Global Positioning System

Header data: A specific set of metadata that consists of essential information to decode the image data (such as the image size, bit depth, colorspace and compression method, etc.) and additional information required to locate the image data and the other metadata within a file.

ICC: International Color Consortium

Image Creation: Information on the creation of the image. This includes the original capture, plus information on when the image is originally stored as a digital image.

Image data: A digitally encoded representation of a picture that is usually compressed but may also be stored as uncompressed raw signal data of the capture device.

Intellectual Property Rights: Information on the owner of an image. This information includes not only the producer of the photograph, but also information on anyone with an interest in the image.

Interoperability: Independent systems that communicate are said to be interoperable. This interoperability is usually performed by both systems using a similar data model.

IPMP: Intellectual Property Management and Protection

IPR: Intellectual Property Rights

JFIF: JPEG File Interchange Format

JPEG: Joint Photographic Expert Group

JURA: JPEG Utilities Registration Authority

Metadata: Additional information associated with the primary data. It is additional data linked with the image data beyond the pixels that define the image in the context of this specification.

Metadata Association: Metadata, by definition is data about data. The data and metadata may be stored together or separately. The metadata is associated with the data either by proximity (they are stored in the same file), or by reference. Where a reference is used, keeping the reference valid requires processing. See Appendix II for details.

MIME: Multipurpose Internet Mail Extensions

NMEA: National Marine Educators Association

Organization: A defined group, whether officially defined such as a commercial company, or informal such as a sporting team or a family.

PAR: Print Aspect Ratio

Private Metadata: Metadata used within a closed system. This may be stored in a file that is primarily interoperable, but the fields and their meaning are not designed to be universally interpreted.

Public metadata: Metadata defined to be interoperable. The format and meaning of the metadata is clearly defined, and expected to be used by multiple systems.

SPIFF: Still Picture Image File Format

SVG: Scalable Vector Graphics

TIFF: Tagged Image File Format

Timestamp: An exact time of an event. This is used throughout this specification to define when metadata was created or modified. See Section 5.6 for the usage.

TTP: Trusted Third Party

UID: Unique Identifier

URI : Universal Resource Identifier

URL: Universal Resource Locator

UTF: UCS transformation formats

UUID: Universal Unique Identifier

WAP: Wireless Access Protocol

WIPO: World Intellectual Property Organization

XML: Extensible Markup Language

Appendix VIII: DIG35 Working Group

This document has been prepared by the DIG35 Initiative Group (IG) of the Digital Imaging Group. The IG includes the following individuals (in alphabetical order):

Mo Asagari, PhotoChannel Networks; Jean Barda, NETIMAGE; Craig Brown, Canon; Howard Bussey, Eastman Kodak; Troy Chinen, Fuji Photo Film; Sheldon Fisher, Microsoft; Scott Foshee, Adobe; Kats Ishii, Canon *(Chair and editor)*; Minoru Ishikawa, Fuji Photo Film; Kiyoshi Kusama, Canon; Tim Long, Canon; George Lynch, Hewlett Parkard; Kentaro Matsumoto, Canon; Rob Reisch, Eastman Kodak; Robert Schultz, WorkStation.com; Dan Sanow, PhotoWorks, Inc.; George Sotak, Eastman Kodak; David Wilkins, PictureIQ; Warren Whaley, Canon; Beverly Valdez, GaiaTech Inc.

Appendix IX: References

[1] Adobe Systems. *TIFF Revision 6.0 Final*. June 3, 1992

[2] ASTM. *Standard Practice for Electronic Interchange of Color and Appearance Data.* E1708-95. 1995

[3] DIG. *Flashpix format specification*. Version 1.0.1. July 1997

[4] ICC. *ICC profile format specification.* ICC.1:1998–09

[5] IETF. *Key words for use in RFCs to Indicate Requirement Levels.*RFC 2119. March 1997

[6] IETF. *Tags for Identification of Languages*. RFC 1766. March 1995

[7] IETF. *Uniform Resource Identifiers (URI)*. RFC 2396. August 1998

[8] IETF. *UTF–8, A transformation format of ISO 10646.* RFC 2279. January 1998.

[9] IETF. *vCard MIME Directory Profile*. RFC 2426. September 1998

[10] ISO. *Codes for the representation of names of countries and their subdivisions – PART 1 Country Codes.* ISO 3166-1. October 1997

[11] ISO. *Codes for the representation of names of countries and their subdivisions – PART 2 Country subdivision code.* ISO 3166-2. December 1998

[12] ISO. *Codes for the representation of names of countries and their subdivisions – PART 3: Code for formerly used names of countries.* ISO 3166-3. March 1999

[13] ISO. *Data elements and interchange formats – Information Interchange – Representation of dates and times*. ISO 8601. 1998

[14] ISO. *Photography – Electronic still picture cameras – Determination of ISO speed*. ISO 12232:1998.

[15] ISO. *Photography – Electronic still picture cameras – Resolution measurements*. ISO/DIS 12233.

[16] ISO. *Photography – Electronic still picture imaging – Removable memory – Part 2: Image data format – TIFF/EP*. ISO/DIS 12234-2

[17] ISO. *Photography – Electronic still picture cameras – Methods for measuring opto-electronic conversion functions (OECF's)*. ISO/DIS 14524.

[18] ISO. *Information technology – Digital compression and coding of continuous-tone still images: Requirements and guidelines*. ISO/IEC 10918-1:1994.

[19] ISO. *Information technology – Digital compression and coding of continuous-tone still images: Registration of JPEG Profiles, SPIFF Profiles, SPIFF Tags, SPIFF colour Spaces, APPn Markers, SPIFF, Compression types and Registration authorities (REGAUT)*. ISO/IEC 10918-4

[20] ISO. *Information technology -- JPEG2000 Image Coding System.* ISO/IEC FDIS 15444-1:2000.

[21] ISO. *Information Technology — Open Systems Interconnection — Remote Procedure Call.* ISO/IEC 11578:1996.

[22] JEIDA. *Digital Still Camera File Format Standard (Exif)*. Version 2.1. June 1998

[23] John S Denker. *See How It Files* 1996

[24] NMEA. *NMEA 0183: Standard For Interfacing Marine Electronic Devices.* Version 2.30. March 1998

[25] WIPO. *Berne Convention for the Protection of Literary and Artistic Works*. Paris Act of July 24. 1971. amended September 28. 1979.

[26] WIPO. *World Intellectual Property Organization Copyright Treaty*. 1996.

[27] W3C. *Date and Time Formats*. TR-NOTE-datetime. September 1997

[28] W3C. *Extensible Markup Language (XML 1.0).* Rec-xml-19980210. Feburary 1998

[29] W3C. *Namespaces in XML*. Rec-xml-names-19990114. Janurary 1999

[30] W3C. *XML Schema Part 0: Primer*. WD-xmlschema-0-20000407. April 2000

[31] W3C. *XML Schema Part 1: Structures*. WD-xmlschema-1-20000407. April 2000

[32] W3C. *XML Schema Part 2: Datatypes*. WD-xmlschema-2-20000407. April 2000

Index