INTERNATIONAL STANDARD

ISO/IEC 19566-8

First edition
2023-02

Information technologies — JPEG systems —

Part 8: JPEG Snack

Technologies de l'information — Systèmes JPEG —

Partie 8: JPEG Snack *définissant des métadonnées d'enrichissement destinées à faciliter la consommation des contenus JPEG*

Reference number
ISO/IEC 19566-8:2023(E)

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

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

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

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

A list of all parts in the ISO/IEC 19566 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

The ISO/IEC 19566 series, on JPEG systems, contributes to the specification of system-level functionalities.

JPEG Snack is a means to convey relatively simple multimedia experiences which is fundamentally based on images and the image file format. Many digital storytelling experiences are based on converting images into video-based technologies, whereas images are directly used in JPEG Snack, along with playback of other media (video, audio, titles, captions, and effects) coordinated through an explicit timeline.

Information technologies — JPEG systems —

Part 8:
JPEG Snack

1 Scope

This document defines JPEG Snack metadata that enriches a representation of multiple media contents, in order to facilitate sharing, editing, and presentation; it further specifies metadata and container formats for JPEG Snack format.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 10918-1, *Information technology — Digital compression and coding of continuous-tone still images: Requirements and guidelines*

ISO/IEC 15444-2, *Information technology — Part 2: Extensions*

ISO/IEC 18477-3, *Information technology — Scalable compression and coding of continuous-tone still images — Part 3: Box file format*

ISO/IEC 19566-5, *Information technology — Part 5: JPEG Universal Metadata Box Format (JUMBF)*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 10918-1 and ISO/IEC 18477-3 and the following apply.

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

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

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

3.1
snack culture
consumption of image-rich media in a short story format

3.2
media type
indicator of the format and content of the file transmitted through the Internet.

3.3
z-order
ordering of overlapping two dimensional regions that define the occlusion precedence amongst them

4 Overview

This document specifies metadata and formats that enable storing, sharing, and rendering snack culture contents with JPEG image coding standards.

NOTE The snack culture contents are defined as follows:

— image sequence from which one or more frames are generated by manipulating still images;

— image sequence recorded with a short playing duration, e.g. 1.5 s;

— image sequence with transition effects and/or overlay along with subtitles, audio clips, and graphics.

JPEG Snack is a format that defines the representation of multimedia, such as images, image sequences, text, audio, and video clips, including transition effects, based on the existing JPEG family image coding standards. Besides, it supports a timing mechanism to synchronize multimedia with a global timeline in a context. This mechanism allows users to watch multimedia contents like short-form video clips. However, unlike conventional video formats, it supports storing images without transcoding from image to dedicated video codec.

In order to define the functionalities of the JPEG Snack format, this document is organized as follows:

— 4.1 describes the overall system of the JPEG Snack format.

— 4.2 describes the system decoder model.

— 4.3 defines an essential model of metadata to compose the JPEG Snack format.

— Clauses 5 and 6 describe the JPEG Snack format in detail.

— Annexes A to C explain how the metadata is serialized and describe the formation of the JPEG Snack file and its usage examples.

4.1 System description

This document specifies metadata and its behaviour to compose the JPEG Snack content by synchronizing multimedia on the decoder side. This document primarily defines a metadata model consisting of two formats:

— Object-structured format: describes the content and additional behaviours of the objects are structured in the object-composition description.

— Object-composition format: describes the positional and temporal relationships between objects and the composition of the objects onto the decoder display.

Its hierarchical structure of the JPEG Snack format is depicted in Figure 4.1.

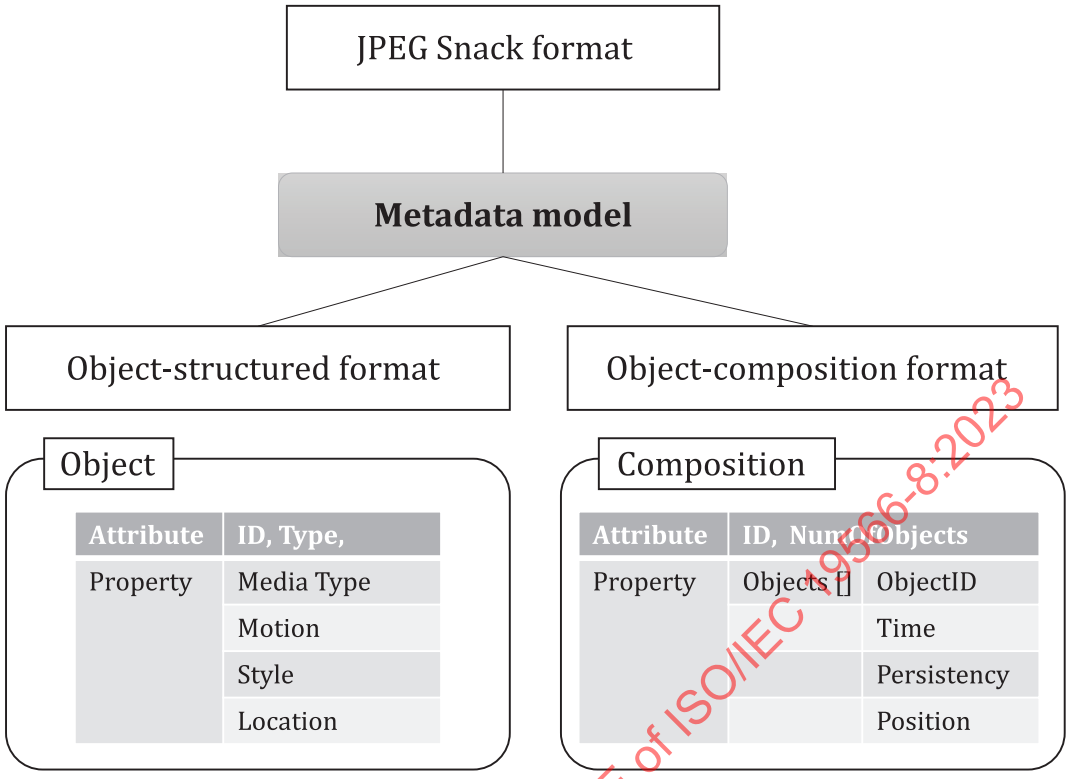


Figure 4.1 — Overview of the JPEG Snack format

The JPEG Snack format provides information that enables JPEG Snack applications to share and render media contents by accessing the objects in the file or reference to objects contained in other files. All objects are not necessarily embedded in the same file. Each object constituting a JPEG Snack file is structured using a box defined in ISO/IEC 19566 and stored into a JPEG image file.

The object-structured format defines the appearance and behaviour of the individual object. This format includes the size and opacity of the object, movement information in a given timeline of the representation, and information on the location where the media data, such as an image codestream, is found (see Clause 5).

The object-composition format identifies the objects that compose the representation and defines each object's creation and destruction. This format describes the temporal and spatial relationship between objects by providing information on the time and position of the individual object to show, and the time and position of their disappearance. Each object has independent position information on the decoder screen, and the composition information determines the z-order of the objects displayed to the user (see Clause 6).

4.2 System decoder model

A JPEG Snack decoder implements the metadata model described in 4.1. The decoder has three conceptual necessary components: default image, timeline, and layer and position, as depicted in Figure 4.2. The decoder decodes the JPEG image to prepare a default image and compose a JPEG Snack representation with several objects using this default image as a background. Since the JPEG Snack is created by defining when, where and how objects are composed, the decoder shall handle timeline, layer, and position.

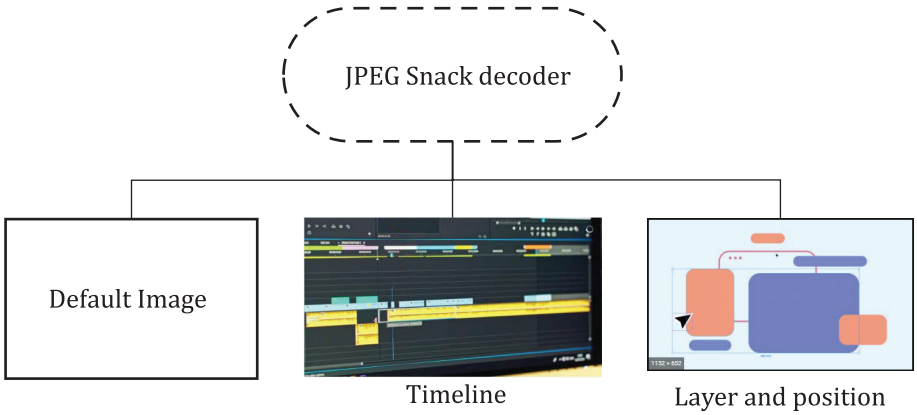
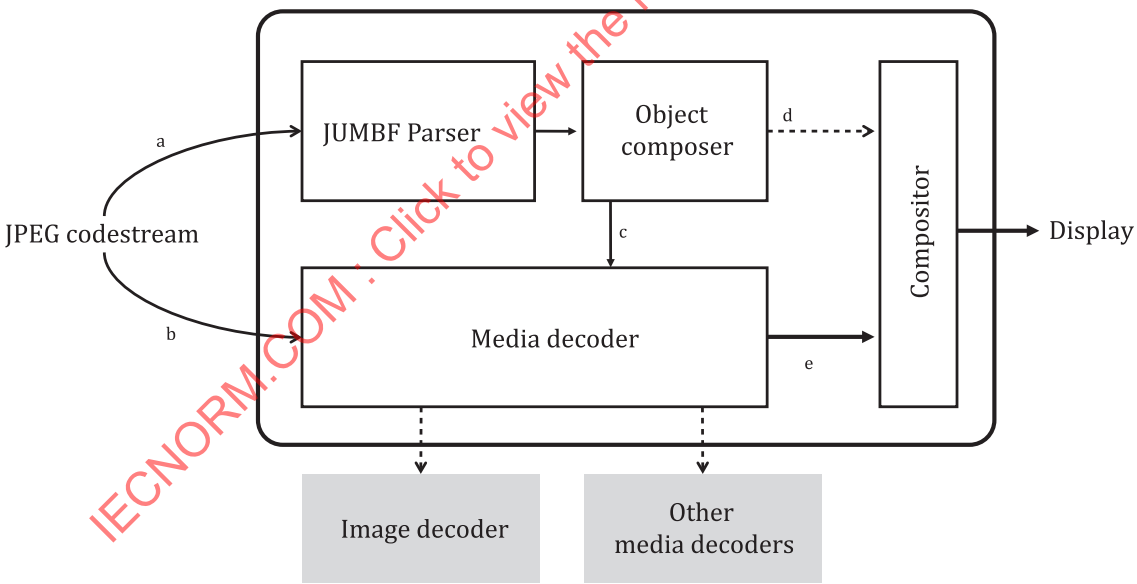


Figure 4.2 — Overview of the JPEG Snack decoder

This document defines the formats based on the informative system decoder model of JPEG Snack, as depicted in Figure 4.3, to allow various JPEG image coding standards to represent JPEG Snack contents in a concerted way. Figure 4.3 illustrates an example of the JPEG Snack decoder in which the formats defined in 4.1 may be implemented.

In Figure 4.3, the object composer receives a JPEG codestream that contains metadata and media data through the JUMBF parser, constructs the JPEG Snack representation, invokes media decoders to decode its media data from the codestream, and renders the media content decoded to the output devices. The object composer controls the media decoder and compositor to decode and display its media content regarding time and position appropriately. This version of the document allows images, captions, image sequences, audio clips, video clips to be composed in a representation of JPEG Snack.



a Metadata.
b Media data.
c Media format + time.
d Position + z-order.
e Media output.

Figure 4.3 — Overview of the system decoder model for JPEG Snack

4.3 Metadata model

The system decoder model described in 4.2 is based on the JPEG Snack format depicted in Figure 4.1 to support the playback of JPEG Snack contents being constituted by multiple media contents.

The metadata is a hierarchical model, as illustrated in Figure 4.4, containing multiple object metadata (see Clause 6) aligned with composition metadata corresponded to the object-composition format. Within the object metadata corresponded to the object-structured format, properties (see Annex A) composing the objects into a representation of the JPEG Snack format such as position, time, and transition are contained. Each object may be rendered individually in a logical timeline of the decoder to support re-editing the object; for example, a user may choose a specific object to hide in his/her JPEG Snack viewer.

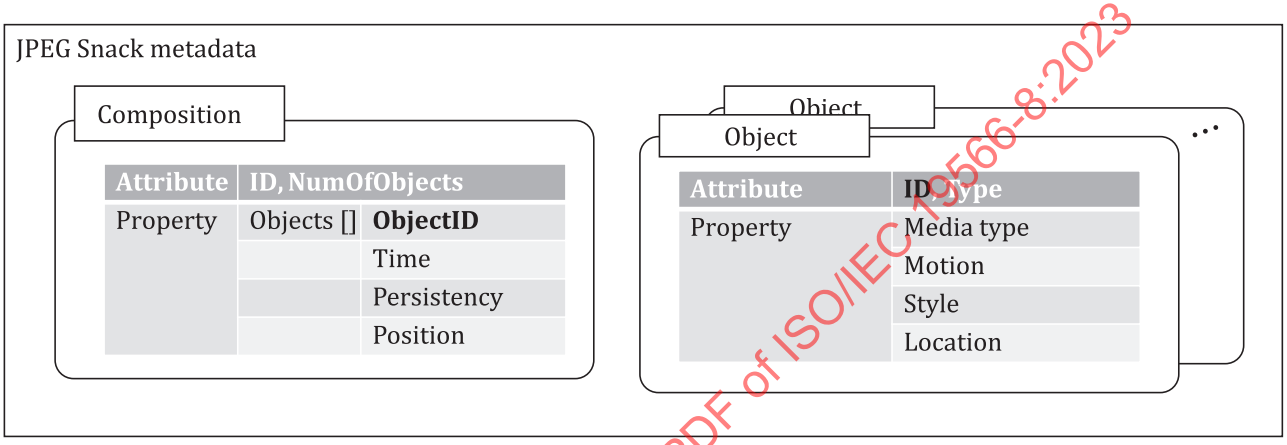


Figure 4.4 — High-level metadata model of JPEG Snack

Object metadata specifies the content and additional behaviour of the individual objects that compose the representation and identifies where the object's resides. An ID is an identifier of the object in the representation and a Type attribute allows a decoder to recognize properties of the object proactively.

Composition metadata coordinates the objects composing a JPEG Snack representation. The objects are arranged into Objects within a composition along with position and time with an identifier attribute. A Position property determines where the object pointed to by the ObjectID is placed. When objects are overlapped according to the Position property, the Time and Persistency properties organize the objects to be placed in front or behind the other object (see 6.2).

JPEG Snack shall have only one composition metadata consisting of one or more objects within a scope of the JPEG Snack file.

The JPEG Snack decoder described in 4.2 composes a timeline (see 6.1.2) for playback of the JPEG Snack content by combining the Time information of all objects, and they exist in the representation individually using their Position and Time information.

4.4 Object-structured file organization

An object in the file organization is a JUMBF box. The JPEG Snack files are formed as a series of boxes. All metadata is contained in boxes, as illustrated in Figure 4.5. JUMBF boxes for JPEG Snack contains metadata to compose the JPEG Snack representation, and other types of JUMBF box are used to deliver the media content, such as a codestream and XML document for each object. The boxes shall be embedded as defined in Annex A and ISO/IEC 19566-5.

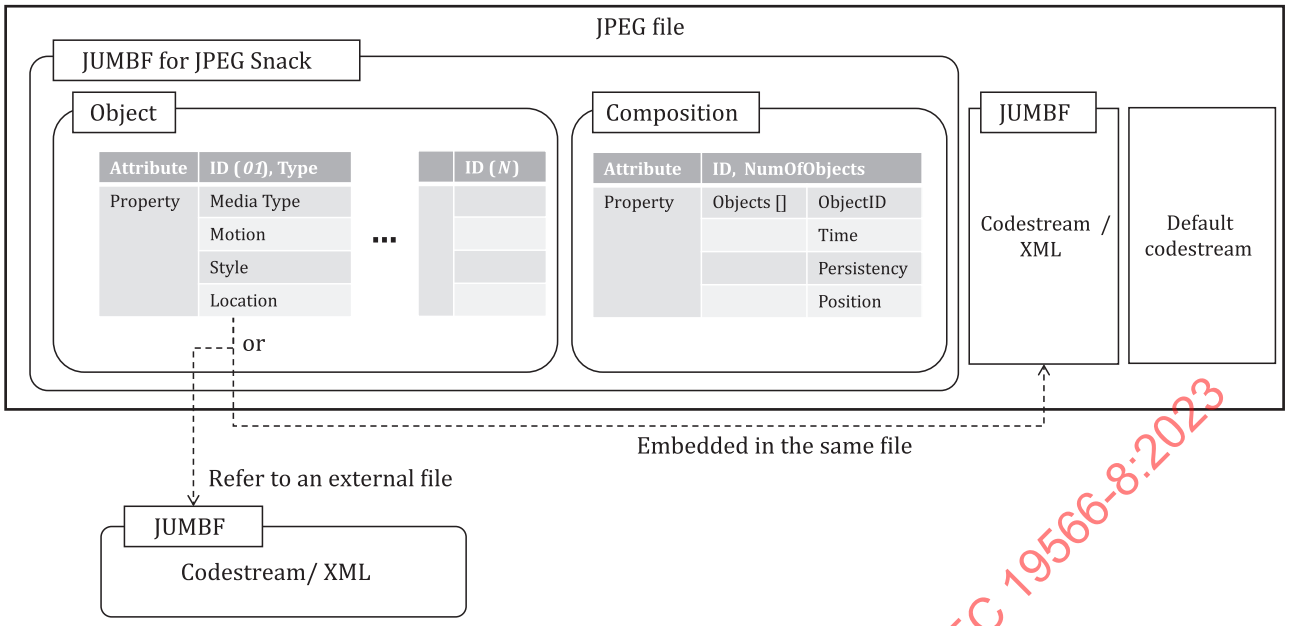


Figure 4.5 — Organization of the JPEG Snack file

The JPEG Snack format provides information to define the metadata for composing the representation and the format in which the metadata is structured in the JPEG image files. The JPEG Snack file has a different file extension according to the default codestream. Conventional JPEG decoders may ignore JUMBF boxes for the JPEG. For example, if the JPEG Snack metadata is embedded in the file of the ISO/IEC 10918-1, denoted by JPEG-1, the extension of the JPEG Snack file is '.jpg' like conventional JPEG-1 images while the conventional JPEG-1 decoder decodes only the default codestream. This feature provides compatibility to the existing JPEG image coding standards, including future standards based on the box-based format.

NOTE 1 The default codestream is placed at the end of the file to be compatible with the conventional JPEG image coding standards. For example, the JPEG-1 decoder can ignore any extra data beyond the EOI (end of image) marker.

NOTE 2 Codestream is a sequence of bits representing a compressed image and associated metadata.

In addition, content types of which is indicated by the object metadata may be different JUMBF boxes based on the object type. The object may refer to JUMBF boxes for media data embedded in another file. The referencing shall be done as defined in ISO/IEC 19566-5:2019, Annex C.

5 Object-structured format

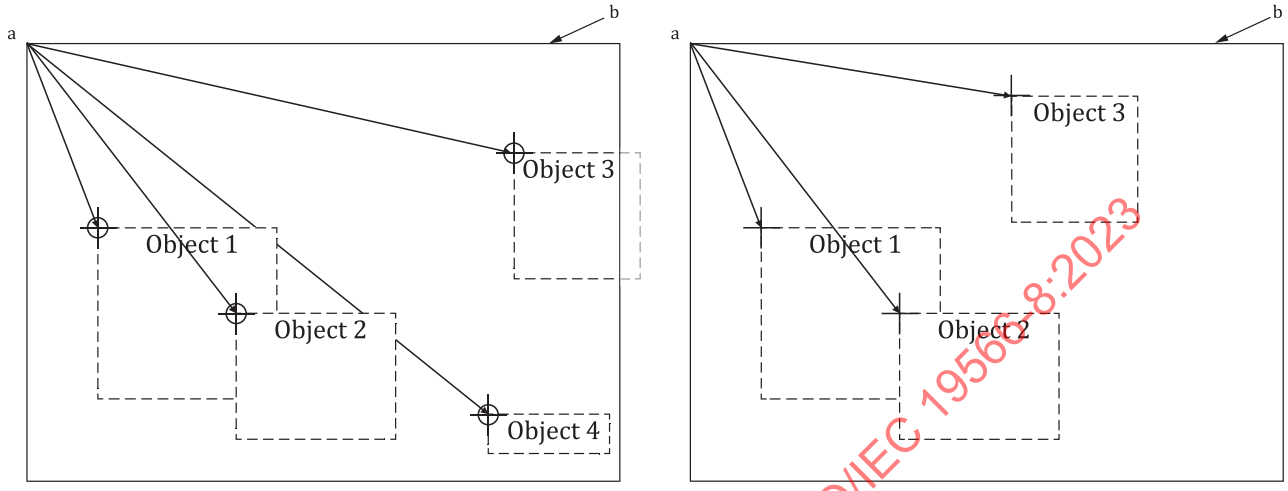
5.1 General

As described in Clause 4, in the JPEG Snack format, the representation of the JPEG Snack is composed of a group of media contents. The object in this document is a unit that composes a JPEG Snack format and contains information to represent the media contents.

Figures 5.1 and 5.2 illustrate the roles of the object-composition and object-structured formats to compose JPEG Snack representation. The object-composition format (see Clause 6) provides composition information to define when and where the objects that are constructed will appear and disappear in a representation, whereas the object-structured format signals information on the individual object's behaviour and location of the resource. In Figure 4.3, while the object composer manages instances of the object, the decoding of the individual object is conducted independently by the media decoder. The

object composer informs the compositor z-order and movement information of the object. Then the compositor renders the decoded media data accordingly based on the z-order and position information.

NOTE An invisible object, such as an audio clip, does not have z-order and position information. And a description of spatial audio is not included in this document, whereas it is considered as a typical audio clip.



a) Composition of objects at t_0 **b) Composition of objects at t_1**

Key

t_0 time when the representation is started

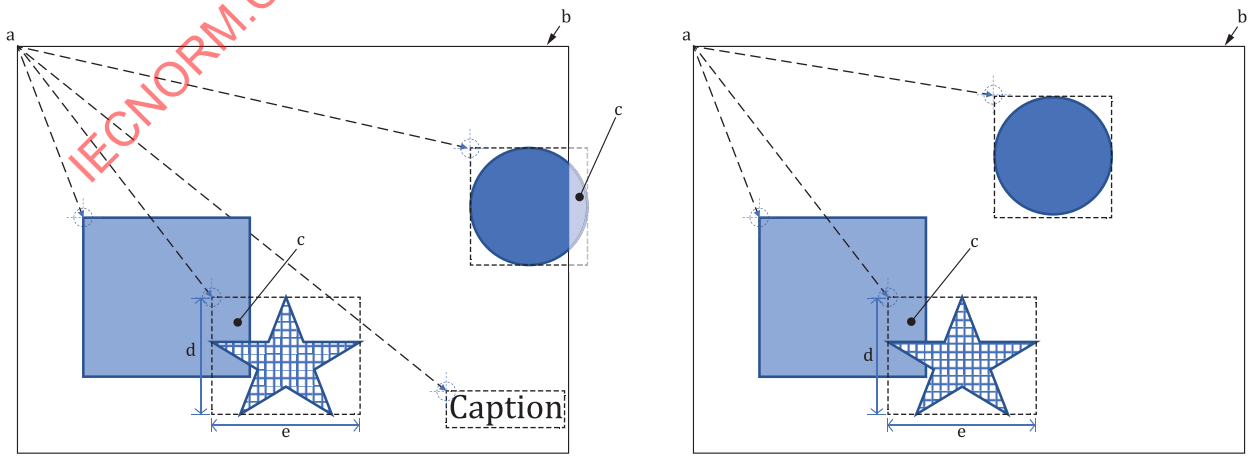
t_1 time when the representation is ended

a Origin.

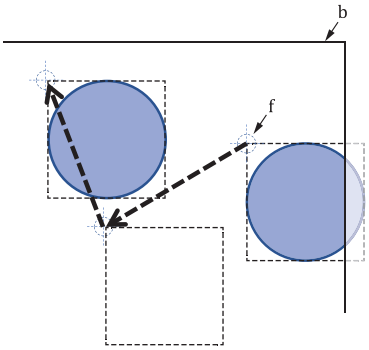
b Representation.

Figure 5.1 — Example of the object-composition format. The t_0 is a time when the representation is started and the t_1 is when the representation is ended.

In the example of Figure 5.1, object 2 is above object 1 so that object 1 has an occluded region. Also, object 3 has an occluded region beyond a representation. The object composer shall handle these regions smoothly. For object 4, the duration of existence is shorter than the JPEG Snack's total duration. See Clause 6.1.2 for more details on the temporal composition of objects.



a) Objects' shape at t_0 **b) Objects' shape at t_1**



c) Object's movement from at t_0 to t_1

a Origin.
b Representation.
c Hidden region.
d Height.
e Width.
f Object's origin.

Figure 5.2 — Example of the object's movement. c) exemplifies a moving object by instruction set.

In Figure 5.2, object 3 moves to another position as time goes from t_0 to t_1. The movement of the object shall be defined by instruction sets as depicted in Figure 5.2 c). Details on the mechanism of moving objects are described in subclause 6.2.4.

NOTE Objects 1, 2, 3, and 4 in Figure 5.1 correspond to the rectangular, star, circle, and caption in Figure 5.2, respectively.

5.2 Object definition

5.2.1 General

This subclause defines the object-structured format that specifies the semantics of the objects that compose a JPEG Snack representation, and the syntax is defined in Annex A. Table 5.1 describes the semantics of the object with attributes and elements to define the media content and the object's properties within the representation.

Attributes and elements define the shape and behaviour of the media content, which is an object that makes up the representation, and determine the object type as listed in Table 5.1 (see 5.2.3 and 5.2.4). Attributes contain media content information that can identify an object, and elements define properties for rendering the object on the representation. Table 5.1 describes the meaning of the parameters constituting an object.

Table 5.1 — Semantics of the object

Attribute name	**Description**
Id	An identifier for this object which is a non-zero 8-bit integer. This shall be unique in a JPEG Snack file.
Type	A string in UTF-8 characters. Either 'static' or 'dynamic' shall be defined.
Number of media	An integer declares a number of the media content corresponding to this object. In the case of the dynamic object, for example, consecutive image sequences shall be identified in the scope of a single object.
Media type	An identifier for the media contents of the object. When the number of media is greater than 1, the corresponding media contents shall be the same media type.
Element name	**Description**

Table 5.1 *(continued)*

Style	An identifier provides a unique address where a resource can be found. For more details see ISO/IEC 19566-5. The resource provides an additional style of the object, such as transition and font-family. For more details see Cascading Style Sheets (CSS) specification[1].
Opacity	A floating-point number range 0–1 that provides a condition of being transparent. 1 means that this object is fully opaque.
Location	An identifier provides a unique address where a resource can be found. For more details see ISO/IEC 19566-5.

5.2.2 Object types and media types

The type of the object shall be determined by the media type of the content. This document is for the object types listed in Table 5.2, and the content type of the object is designated using media types, as described in Table 5.2. Definitions of the object type are also provided in the table.

Table 5.2 — Supported media types

Object type	Media type	Description
Image	All media of type image as listed in the IANA Media Type Registry[1]	A still image. See 5.2.3.1.
Caption	text/markdown	A piece of text for additional information. See 5.2.3.2.
Pointer	All media of type image as listed in the IANA Media Type Registry[1]	A graphical indicator used to arouse attention to the region of interest that works as a presentation pointer. See 5.2.3.3.
Image sequence	All media of type image as listed in the IANA Media Type Registry[1]	A set of consecutive still images. See 5.2.4.1.
Video clip	All media of type video as listed in the IANA Media Type Registry[1]	A briefly recorded file used to convey audio-visual information to the audience. See 5.2.4.2.
Audio clip	All media of type audio as listed in the IANA Media Type Registry[1]	A briefly recorded file used to convey audible information to the audience. See 5.2.4.3.

When a JPEG Snack decoder does not support a media type, the corresponding object is ignored, and the decoder shall inform that the object is missing in the representation.

JPEG Snack differentiates image and image sequence while both objects use the same media type. Even though several images are contained in a single JPEG Snack representation, those images are relatively less correlated in the context of the representation, which means that occlusion or exclusion of some images out of several images do not harm the representation. However, images in a sequence are highly correlated to create a valid context of the representation. For example, when one of the frames in animation is skipped, the animation may look strange.

In this document, objects are categorized into static and dynamic objects based on if the object's contents are changing during the representation as time goes. Details on the object are defined in 5.2.3 and 5.2.4.

5.2.3 Static objects

This version of the document defines image, caption, and pointer as static objects. The value of the type attribute shall be a 'static' and number of media shall be 1, while other attributes and elements vary

to the content of the object, as described in Table 5.3. Style and opacity elements are optional. If those elements are absent, the object has a fixed position and fully opaque in the representation.

Table 5.3 — Semantics of the static object

Attribute name	**Value**	**Description**
Id	1...N	Shall be provided.
Type	'static'	
Number of media	1	
Media type	string	Shall be provided.
Element name	**Value**	**Description**
Style	string	Optional.
Opacity	0...1	Optional. Default value is 1.
Location	string	Shall be provided.

5.2.3.1 Image

The object mainly used in the JPEG Snack format is a still image, and any image format can be used as defined in Table 5.2. The JPEG Snack format may contain multiple images. In this case, images taken over a relatively long-term period are mainly used, and images may be selectively added or removed according to the user's selection as an individual object. Table 5.4 defines default values for the image object. Number of media element shall be 1 while others are varied to the content of the object. Size information may be different to the resolution of the image, and the image shall be scaled and rendered with the size of the object. If an aspect ratio of the resolution is different from the object size, the ratio shall be kept.

Table 5.4 — Default values for the image object

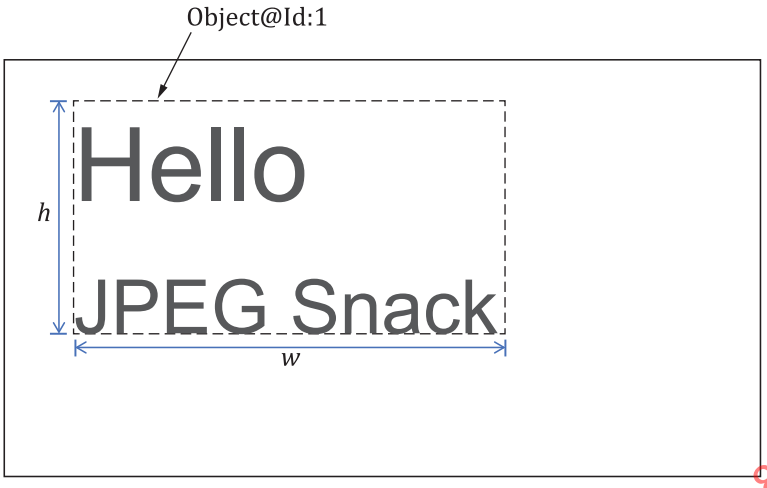
Attribute name	**Value**
Id	Shall be provided.
Type	static
Number of media	1
Media type	Shall be provided.
Element name	**Value**
Style	Optional
Opacity	Optional
Location	Shall be provided.

In contrast to the image, the image sequence (see 5.2.4.1) is a group of continuous images captured for a few seconds, and when a part of the continuous images is removed or added, the representation of the JPEG Snack becomes unnatural.

NOTE This document defines formats that support an animation, such as GIF format, as static objects.

5.2.3.2 Caption

The caption is Markdown, which is a piece of plain-text writing, that is overlaid at a specific position in the representation of the JPEG Snack. This document defines only the size of the object as depicted in Figure 5.3, and formatting syntaxes are defined in RFC 7763.[2] The font information is signalled by the style element in the object metadata and as defined in CSS specification.[3]



Key

w width of the object

h height of the object

Figure 5.3 — Example of the caption objects

Table 5.5 exemplifies values for caption objects of Figure 5.3. In the example, the caption is a string with Markdown, as listed in Table 5.6. The size element provides width and height information of the caption to be overlaid at a given position by object-composition format. Motion, transition, and opacity elements are not defined for this example, so the caption is rigid during their existences.

Table 5.5 — Values for the caption object in Figure 5.3

Attribute name	**Value**
Id	1
Type	static
Number of media	1
Media type	text/markdown;charset=UTF-8
Element name	**Value**
Style	Absent
Opacity	Absent
Location	Shall be provided

Table 5.6 — Markdown for the caption object in Figure 5.3

Markdown	**Rendered output**	**Description**
# Hello ### JPEG Snack	Hello JPEG Snack	The number sign (#) in front of a word or phrase creates a heading. — #: Heading level 1 — ###: Heading level 3 See RFC 7763 for the syntaxes.

NOTE The size of the caption object is the size of the border in the CSS box model.[4]

5.2.3.3 Pointer

The pointer is an item of graphics overlaid at a specific position in the JPEG Snack representation. It is used to deliver information quickly and accurately by attracting the viewer's attention.

Figure 5.4 provides an example representing a pointer along with its explanatory caption. The pointer graphically indicates where an author wants to stress by pointing in the default image. The caption object annotates the indication. Table 5.7 lists values for the pointer object of the example. In the example, the location element is an identifier of the image data in PNG (portable network graphics) format.

Key

w width of the object

h height of the object

Figure 5.4 — Example of the pointer object

Table 5.7 — Values for the pointer in Figure 5.4

Attribute name	**Value**
Id	3
Type	static
Number of media	1
Media type	image/png
Element name	**Value**
Style	Absent
Opacity	Absent
Location	Shall be provided

5.2.4 Dynamic objects

This version of the document defines image sequence, video clip, and audio clip as dynamic objects. The value of the type attribute shall be a 'dynamic', while other attributes and elements vary to the content of the object, as described in Table 5.8.

Table 5.8 — Semantics of the dynamic object

Attribute name	Value	Description
Id	1...N	Shall be provided.
Type	'dynamic'	
Number of media	1...N	Shall be provided.
Media type	string	Shall be provided.
Element name	**Value**	**Description**
Style	string	Optional.
Opacity	0...1	Optional. Default value is 0.
Location[]	A series of locations	Shall be provided.

5.2.4.1 Image sequence

Although an image object belongs to the static object as described in 5.2.3.1, the JPEG Snack format defines an object consisted of multiple images, which is an image sequence, as a single dynamic object and all images shall be the same media type. Table 5.9 describes attributes and elements for the image sequence object and the number of media shall be larger than 1.

Table 5.9 — Values for the image sequence object

Attribute name	Value
Id	Shall be provided
Type	dynamic
Number of media	Shall be larger than 1
Media type	All image types
Element name	**Value**
Style	Optional
Opacity	Optional
Location	Shall be provided

The style and opacity shall be applied at the scope of the object, not at the scope of the components contained within the object.

5.2.4.2 Video clip

A video clip is an object that works like the image sequence, whereas the number of media and media type are different from it. The number of media shall be 1. If the length of the video clip is less than the value of the LIFE parameter of the object-composition format (see 6.2), then the playback shall restart from the beginning of the clip.

5.2.4.3 Audio clip

Audio clip is an invisible object in the representation. Style and opacity elements are not used as defined in Table 5.10.

The audio clip may consist of several audio data. If the length of audio clips is less than the value of the LIFE parameter of the object-composition format (see 6.2), then the playback shall reset from the first audio data.

Table 5.10 — Values for the audio clip object

Attribute name	Value
Id	Shall be provided

Table 5.10 *(continued)*

Type	dynamic
Number of media	Shall be provided
Media type	See 5.2.2
Element name	**Value**
Style	Absent
Opacity	Absent
Location	Shall be provided

6 Object-composition format

6.1 General

A JPEG Snack file contains a single media or multiple media. The object-composition format in the metadata model describes how media contents are composed and synchronized in the representation in accordance with relationships between contents. The composition information of objects is embedded in the file that contains metadata required by a JPEG Snack system decoder to compose and render media contents.

As defined in the system decoder model (see 4.2), two essential components, default image and a timeline, shall be required to compose a JPEG Snack representation. This clause defines those components and the composition of the representation based on the components.

6.1.1 Default image

The JPEG Snack format specifies metadata and its behaviour for JPEG family image coding standards and additional media contents. As illustrated in Figure 4.5, the format shall be embedded in the JPEG image file format. This document defines that the JPEG image is the default image to compose the JPEG Snack representation. The representation shall be built on this default image.

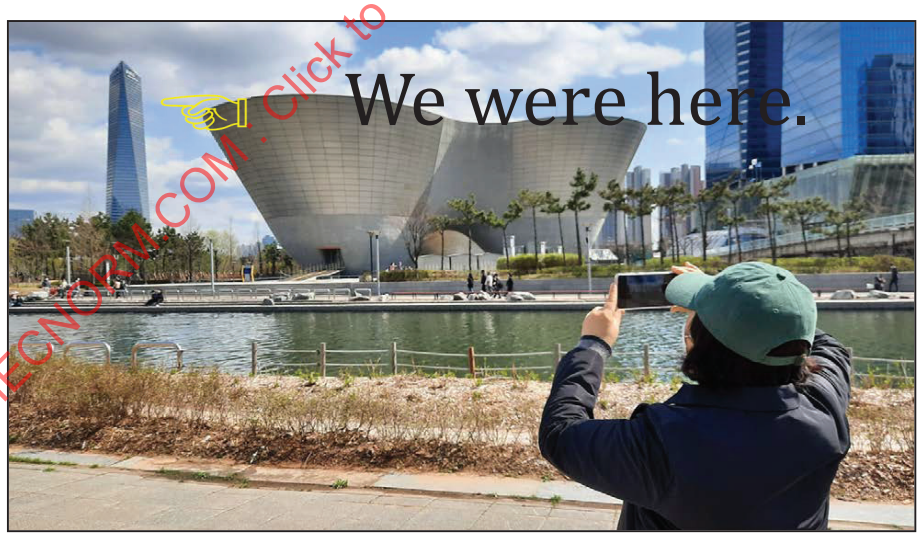


Figure 6.1 — Example of the JPEG Snack: default image and corresponding media contents

Figure 6.1 exemplifies a JPEG Snack file. In the example, the default image is a decoded image of the default codestream as depicted in Figure 4.5 and has the lowest z-order in the representation.

6.1.2 Timeline

A JPEG Snack format has a timeline for the representation, which is the basis to playback media contents contained in the file. Each media shall run its content individually according to the life-cycle in a global timeline of the file.

The timeline shall be constructed after parsing object-composition metadata (see Clause 6.2). Figure 6.2 describes the timeline for the example of Figure 5.1 and how all media contents are synchronized in the global timeline.

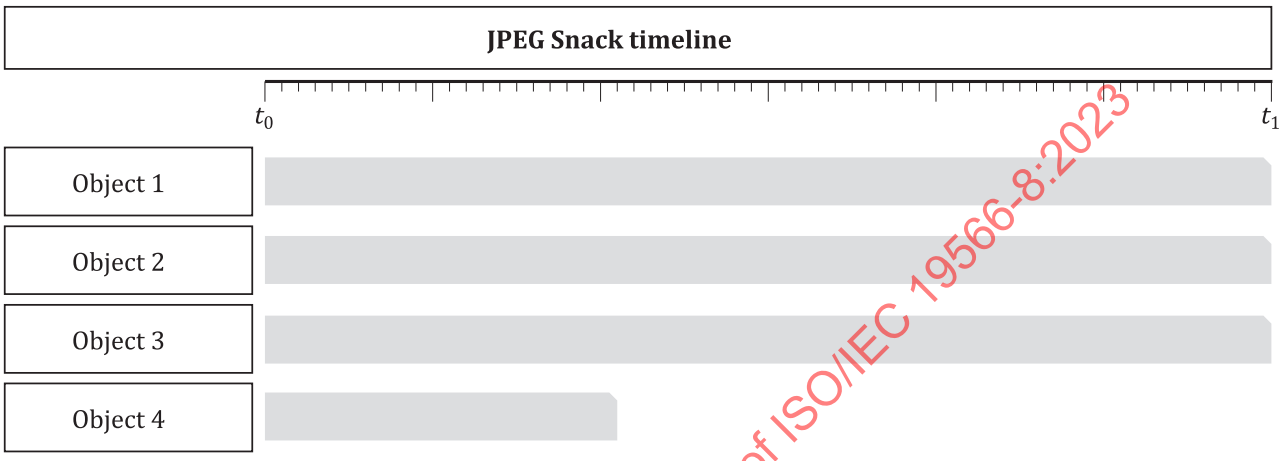


Figure 6.2 — Example of the JPEG Snack timeline corresponding to the example in Figure 5.1

As illustrated in Figure 6.2, object 4 is disappeared while other objects are rendered. Likewise, the start time of objects is not necessarily the same. When a rendering of the representation is finished, the decoder shall provide a function to choose whether the rendering restarts or not.

The timing information of each object shall be constructed based on the corresponding Instruction Set box as defined in the ISO/IEC 15444-2.

6.2 Composing objects

When a system decoder renders a JPEG Snack file, media contents are being composed spatially using position information and temporally using time information, as described in Tables 6.1 and 6.2.

Table 6.1 — Semantics of the composition

Attribute name	**Description**
Start time	A relative time in milliseconds to present this composition on the representation from the moment that the default image is constructed.
Number of composition	Optional. A number of the composition corresponding to this JPEG Snack format. The current version of this document shall contain only one composition.
Element name	**Description**
Composition ID	Optional. An identifier for this composition.
Number of objects	A number of the objects corresponding to this composition.
Object IDs	Identifiers of the objects that composes this composition.

Table 6.2 — Semantics of the instruction set

Parameter	**Description**
Instruction type	A flag specifies the type of instruction, and thus which instruction parameters shall be found within in this set.

Table 6.2 *(continued)*

Repetition		A number of times to repeat this instruction set.
Duration of timer tick		Duration of the timer tick (used by the LIFE instruction parameter) in milliseconds.
Instruction parameters		A series of instruction parameters.
Instruction set		**Description**
Offset	X	Horizontal location at which the top left corner of the default image in pixels, as described in 6.2.2.
	Y	Vertical location at which the top left corner of the default image in pixels, as described in 6.2.2.
Size	Width	Width of the object.
	Height	Height of the object.
Compositing	Persistence	A flag specifies whether the rendered object shall persist on the representation.
	LIFE	Duration of the instruction. This field specifies the number of timer ticks that should occur between completing the execution of the current instruction and completing execution of the next instruction.
	NEXT-USE	A number of instructions that shall be executed before reusing the object.
Cropping		Horizontal and vertical offset and cropped width and height information.
Rotation		An optional rotation and mirroring that is to be applied as last step after cropping and before rescaling and rendering the object to the representation.

The number of objects attribute informs the iteration count of the objects. The Instruction Set boxes shall contain instruction parameters in order of the objects, as illustrated in Figure 6.3. The number of objects indicated by the object IDs shall be the same as the number of sets in the Instruction Sets boxes. The Instruction Set box shall be one in the JPEG Snack Description box.

NOTE This document defines a subset of the compositing mechanism, which is conducted by the Instruction Set box, for the purpose of representing the JPEG Snack format. More details on the compositing mechanism using the Instruction Set box are defined in ISO/IEC 15444-2:2021, Annex M.

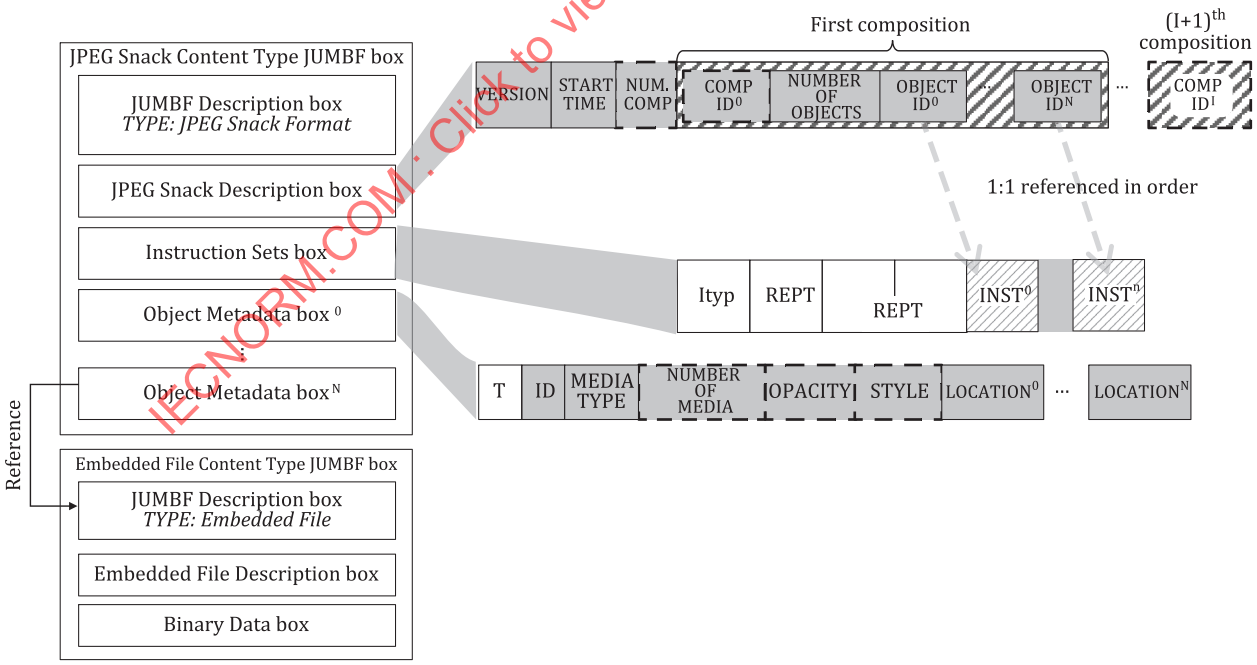


Figure 6.3 — Overview of the JPEG Snack Content Type JUBMF box.

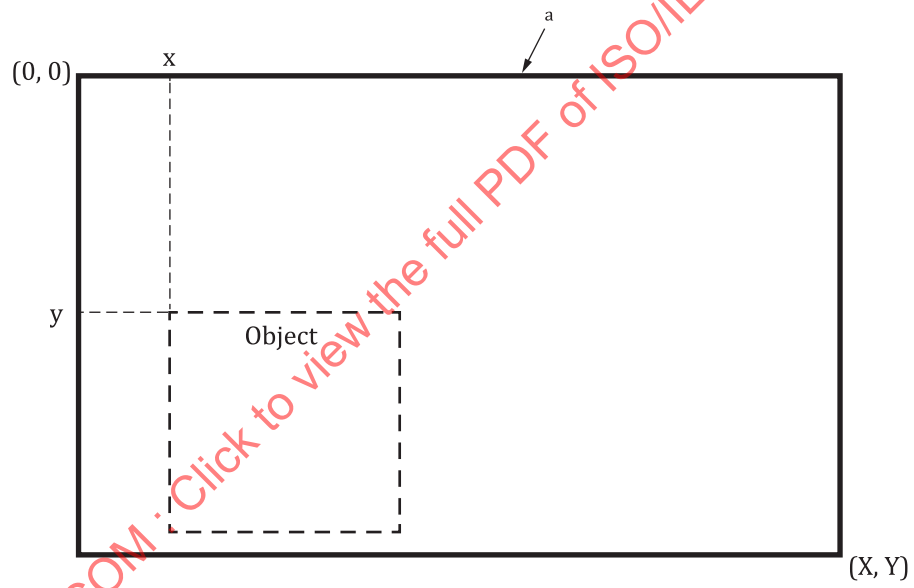
6.2.1 Temporal relationship between the default image and objects

The timeline shall start immediately after decoding and rendering the default image and the whole timeline of the JPEG Snack format is constructed after parsing this object-composition format and the Instruction Set box corresponded. The start time of the composition, as described in Table 6.1, informs the time of when the first object in the composition shall be rendered. All objects in the representation have a lifetime relatively from the start time in the timeline using the persistence and LIFE parameters of the instruction.

The occurrences and disappearance of objects in the timeline shall be determined by the instruction sets as defined in Table 6.2. The LIFE parameter specifies the duration of the object's occurrence in the timeline. In general, the object disappears after the duration is expired unless the persistence is set to 1. These two parameters provide the functionality of rendering multiple objects at the same time.

6.2.2 Spatial relationship between the default image and objects

The representation of JPEG Snack is composed and built onto the default image as a background. The offset parameter of objects corresponding to the object ID layouts the representation by placing the object to where the offset indicates. Figure 6.4 describes how each object is positioned by the offset parameter.



Key

X	width of the default image	a	Default image.
Y	height of the default image	x	horizontal offset
		y	vertical offset

Figure 6.4 — Positioning of the object on the default image

The offset specifies x and y offsets in horizontal and vertical. The size information may be different to the resolution of the image, and the image shall be scaled and rendered with the size of the object. If an aspect ratio of the resolution is different from the object size, the aspect ratio shall be kept.

Although the offset parameter is enough for representing most cases of the JPEG Snack format, additional cropping and rotation may be applied to where the user renders object in a richer representation without additional media data.

6.2.3 Layering the objects

The JPEG Snack format defines a method providing a representation by composing multiple objects. In general, objects are presented at different times for a given duration but can be defined to exist at the same location and time. In this case, this document provides a mechanism for objects to exist hierarchically above or below other objects, as shown in Figure 6.5.

Figure 6.5 — Example of the JPEG Snack: Default image and layered objects

The combinations of persistence and LIFE parameters supports the overlay for captions, pointers, and images. Since the current instruction composes the object on the rendered result of the previous instruction set, the current object identified by the object ID shall be rendered on top of the previously rendered object.

In the example of Figure 6.5, pictures taken at the 'Central park' are presented with corresponding captions. A PIP (picture in picture) image is overlaid on the default image, and the caption object is overlapped on the PIP image. Tables 6.3 and 6.4 show sample values of the example. In this example, pointer, caption 'We were here', PIP image, and caption 'Central park', are identified as objects 1, 2, 3, and 4.

Table 6.3 — Values of the composition format for the example of Figure 6.5

Attribute name	**Value**	**Description**
Start time	0	The composition starts immediately after decoding and rendering the default image.
Element name	**Value**	**Description**
Number of objects	4	This composition contains 4 objects.
Object ID	1	
Object ID	2	
Object ID	3	
Object ID	4	

Table 6.4 — Values of the instruction parameters for the example of Figure 6.5

Parameter	**Value**	**Description**
Instruction type	0000 0000 0000 1011	Each instruction contains offset, size and compositing parameters.
Repetition	0	The instructions are executed once.

Table 6.4 *(continued)*

Parameter	Value	Description
Duration	1000	The timer tick is 1000 ms.
Offset X		As provided.
Offset Y		
Width		As provided.
Height		
Persistence	1	The rendered result of the object persists.
LIFE	0	This instruction is executed within the same display update.
NEXT-USE	0	This instruction is not reused.
Offset X		As provided.
Offset Y		
Width		As provided.
Height		
Persistence	1	The rendered result of the object persists.
LIFE	0	This instruction is executed within the same display update.
NEXT-USE	0	This instruction is not reused.
Offset X		As provided.
Offset Y		
Width		As provided.
Height		
Persistence	1	The rendered result of the object persists.
LIFE	0	This instruction is executed within the same display update.
NEXT-USE	0	This instruction is not reused.
Offset X		As provided.
Offset Y		
Width		As provided.
Height		
Persistence	1	The rendered result of the object persists.
LIFE	2^{31}-1	This instruction delays indefinitely.
NEXT-USE	0	This instruction is not reused.

All objects are presented at the time of when the default image is rendered. A JPEG Snack decoder shall overlay the caption 'Central park' on the PIP image since object 3 includes object 4 geometrically, and the instruction set of object 4 is the last set.

6.2.4 Moving the objects

As described in subclause 5.1, the JPEG Snack format has moving objects. The offset parameter of the instruction set defines the object's movement, as illustrated in Figure 6.6, and the object moved shall be identified by the object-composition format.

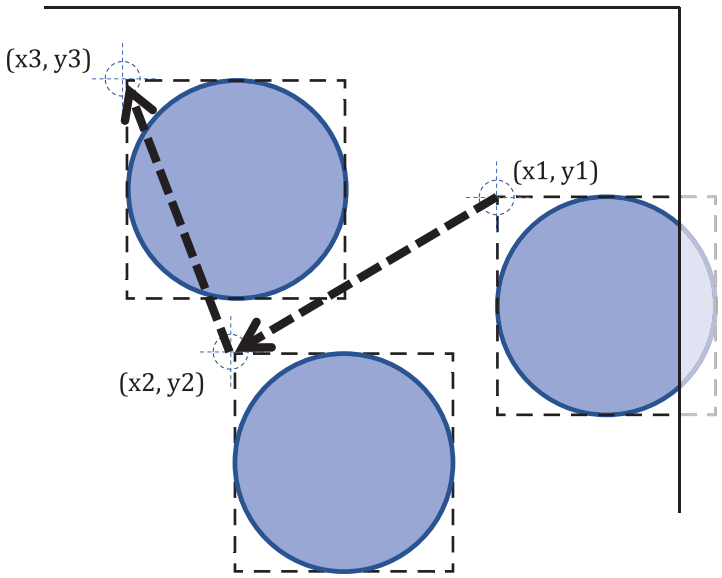


Figure 6.6 — Example of object's movement

The movement shall be specified by applying different offset parameters to the same object. In the example of Figures 5.2 and 6.6, the object is constructed at t_0 and placed at the position (x1, y1). Then it is moved to the position (x3, y3) and destroyed at t_1. As three offset parameters are provided, the object moves to the intermediate position (x2, y2). The rendering duration at each position shall be determined by the LIFE parameter of the instruction. Tables 6.5 and 6.6 show values of the example.

Table 6.5 — Values of the composition format for the example of Figure 6.6

Attribute name	**Value**	**Description**
Start time	0	The composition starts immediately after decoding and rendering the default image.
Element name	**Value**	**Description**
Number of object	3	This composition contains 3 objects. All objects in the composition refer the same media object.
Object ID	1	
Object ID	1	
Object ID	1	

Table 6.6 — Values of the instruction parameters for the example of Figure 6.6

Parameter	**Value**	**Description**
Instruction type	0000 0000 0000 1011	Each instruction contains offset, size and compositing parameters.
Repetition	0	The instructions are executed once.
Duration	1000	The timer tick is 1000 ms.
Offset X	x1	
Offset Y	y1	
Width		As provided
Height		
Persistence	0	The rendered result disappears after 2 s.
LIFE	2	
NEXT-USE	0	This instruction is not reused.

Table 6.6 *(continued)*

Parameter	Value	Description
Offset X	x2	
Offset Y	y2	
Width		As provided.
Height		
Persistence	0	The rendered result disappears after 2 s.
LIFE	2	
NEXT-USE	0	This instruction is not reused.
Offset X	x3	
Offset Y	y3	
Width		As provided.
Height		
Persistence	0	The rendered result disappears after 2 s.
LIFE	2	
NEXT-USE	0	This instruction is not reused.

Annex A
(normative)

Boxes for JPEG Snack

A.1 Overview

This Annex defines boxes of the JPEG Snack format. All values in this annex are encoded as a big-endian unsigned integer.

A.2 JUMBF Content Type for JPEG Snack

A.2.1 JUMBF box Content

JUMBF boxes that embed JPEG Snack metadata shall use the 0x16AD91E0-A37F-11EB-9D0D-0800200C9A66 JUMBF TYPE. The Content of the JUMBF box shall contain exactly one JPEG Snack Description box, exactly one Instruction Set box, and one or more Object Metadata box, as illustrated in Figure A.1.

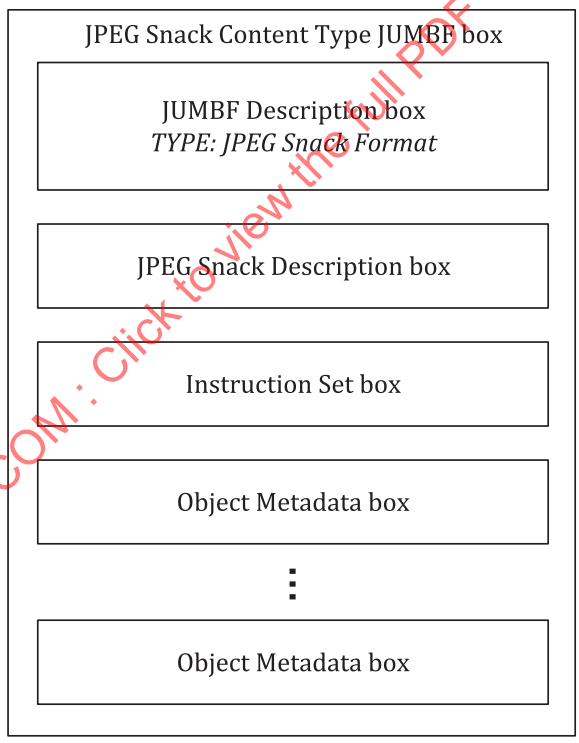


Figure A.1 — Structure of an JPEG Snack Content Type JUMBF box

A.2.2 JPEG Snack Description box

The JPEG Snack Description box signals a number of the objects composing a JPEG Snack representation.

The type of a JPEG Snack Description box shall be 'jsdb' (0x6a73 6462). The contents of the box shall be as in Figure A.2; the fields are summarized in Table A.2.

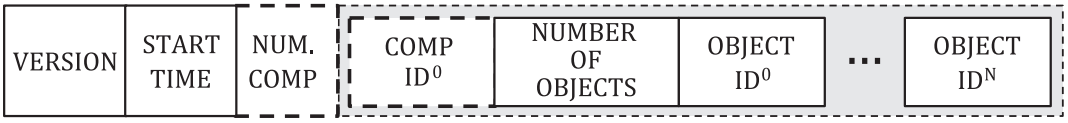


Figure A.2 — Organization of the contents of the JPEG Snack Description box

— VERSION: This field indicates the version of the specification as in Table A.1.

Table A.1 — VERSION field specification

VERSION value	Details
1	This version supports media contents: image, caption, pointer, image sequence, video clip, and audio clip. Media types are supported defined in Table 5.2. This version shall contain only one composition and one Instruction Set box.
All other values are reserved for future use by ISO/IEC.	

— START TIME: This field signals the time in milliseconds to start rendering the composition from the moment that the default image is constructed.

— NUMBER OF COMPOSITION: This optional field signals a number of compositions this format can provide.

— COMPOSITION ID^{I}: This optional field signals an identifier of the Ith composition of this format. This field shall be absent when the NUMBER OF COMPOSITION is not present.

— NUMBER OF OBJECTS: This field signals a number of Object Metadata box corresponding with this box.

— OBJECT ID^{N}: This field signals a set of identifiers corresponding to each composition.

Table A.2 — Format of the contents of the JPEG Snack Description box

Field name	Size (bits)	Value
Version	8	See Table A.1
Start time	64	See Table 6.1
Number of compositions	8	
Composition ID	8	
Number of objects	8	
Object IDs	Variable	

A.2.3 Instruction Set box

The Instruction Set box signals information about the composition of the JPEG Snack representation. The box contains a set of rendering instructions, each represented through a series of composition parameters. In addition, the entire set of instructions contained within this box may be repeated according to a repeat count.

The type of an Instruction Set box shall be 'inst' (0x696E 7374). The contents of the box shall be as in Figure A.3, the fields are summarized in Table A.4.

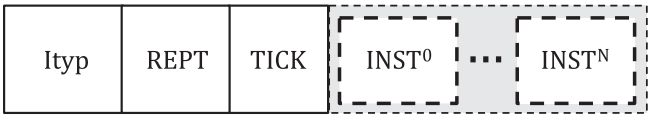


Figure A.3 — Organization of the contents of the Instruction Set box

— Ityp: This field shall contain TOGGLES as in Table A.2. Instruction type. This field specifies the type of this instruction, and thus which instruction parameters shall be found within this JUMBF box. This field is encoded as a 16-bit flag as in Table A.3.

Table A.3 — TOGGLES

Binary value	Meaning
0000 0000 0000 0000	No instructions are present, and thus no instructions are defined for the objects in the file.
0000 0000 0xx0 00x1	Each instruction contains XO and YO parameters.
0000 0000 0xx0 001x	Each instruction contains the WIDTH and HEIGHT parameters.
0000 0000 0xx0 10xx	Each instruction contains the LIFE, NEXT-USE and PERSIST parameters.
0000 0000 0x10 x0xx	Each instruction defines the crop parameters XC, YC, WC and HC.
0000 0000 01x0 x0xx	Each instruction defines the rotation parameter ROT.
	The upper 9 bits are reserved for future use by ISO/IEC.

— REPT: Repetition. The number of times to repeat this set of instructions after executing the instruction set.

— TICK: Duration of timer tick in milliseconds. The tick is defined in the LIFE field of the instruction parameters.

— $INST^N$: Instruction. This field specifies a series of instruction parameters for a single instruction. Iteration count N should be the same as the number of the objects defined in the JPEG Snack Description box. The format of this field is specified in A.2.4.

Table A.4 — Format of the contents of the Instruction Set box

Field name	Size (bits)	Value
Ityp	16	See Table A.3
REPT	16	See Table 6.2
TICK	32	
$INST^N$	Variable	See Table A.5

A.2.4 Instruction parameter

Figure A.4 shows the contents of each individual INST field within an Instruction Set box, and the fields are summarized in Table A.5:

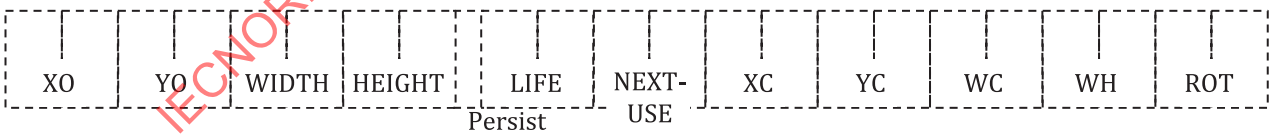


Figure A.4 — Organization of the contents of the INST field within an Instruction Set box

— XO: Horizontal offset. This field specifies the horizontal location at which the top left corner of the object being acted on by this instruction shall be placed in the render area, in samples.

— YO: Vertical offset. This field specifies the vertical location at which the top left corner of the object being acted on by this instruction shall be placed in the render area, in samples.

— WIDTH: Width of the object. This field specifies the width on the render area, in display samples, into which to scale and render the object being acted on by this instruction.

— HEIGHT: Height of the object. This field specifies the height on the render area, in display samples, into which to scale and render the compositing layer being acted on by this instruction. This field is encoded as a 4-byte big endian unsigned integer. If this field is not present, the height of the compositing layer shall be used.

— PERSIST: Persistence. This field specifies whether the object rendered to the display as a result of the execution of the current instruction shall persist on the display background or if the display background shall be reset to the its state before the execution of this instruction, before the execution of the next instruction.

— LIFE: Duration of this instruction. This field specifies the number of timer ticks that should ideally occur between completing the execution of the current instruction and completing execution of the next instruction. A value of zero indicates that the current instruction and the next instruction shall be executed within the same display update; this allows a single frame from the animation to be composed of updates to multiple objects. A value of 2^{31}–1 indicates an indefinite delay or pause for user interaction.

— NEXT-USE: Number of instructions before reuse. This field specifies the number of instructions that shall be executed before reusing the current object. This field allows readers to simply optimize their caching strategy. A value of zero implies that the current image shall not be reused for any ensuing instructions. A value of one (1) implies that the current object will be used with the next instruction and so on. The object passed on for reuse in this manner shall be the original object, prior to any cropping or scaling indicated by the current instruction. If this field is not present, the number of instructions shall be set to zero, indicating that the current compositing layer shall not be reused.

— XC: Horizontal crop offset. This field specifies the horizontal distance in samples to the left edge of the desired portion of the object. The desired portion is cropped from the object and subsequently rendered by the current instruction. If this field is not present, the horizontal crop offset shall be set to 0

— YC: Vertical crop offset. This field specifies the vertical distance in samples to the top edge of the desired portion of the object. The desired portion is cropped from the object and subsequently rendered by the current instruction. If this field is not present, the vertical crop offset shall be set to 0.

— WC: Cropped width. This field specifies the horizontal size in samples of the desired portion of the object. The desired portion is cropped from the object and subsequently rendered by the current instruction. If this field is not present, the cropped width shall be set to the width of the current object.

— HC: Cropped height. This field specifies the vertical size in samples of the desired portion of the object. The desired portion is cropped from the object and subsequently rendered by the current instruction. If this field is not present, the cropped height shall be set to the height of the current object.

— ROT: Rotation. This field specifies an optional rotation and mirroring that is to be applied as last step after cropping and before rescaling and rendering the object to the screen.

This field is encoded as in Table A.5.

Table A.5 — Format of the contents of the INST parameter in the Instruction Set box

Parameter	Size (bits)	Value
XO	32 or 0	0-(2^{32}–1); if Ityp contains xxxx xxxx xxxx xxx1 Not applicable otherwise

Table A.5 *(continued)*

Parameter	Size (bits)	Value
YO	32 or 0	0-(2^{32}–1); if Ityp contains xxxx xxxx xxxx xxx1 Not applicable otherwise
WIDTH	32 or 0	0-(2^{32}–1); if Ityp contains xxxx xxxx xxxx xx1x Not applicable otherwise
HEIGHT	32 or 0	0-(2^{32}–1); if Ityp contains xxxx xxxx xxxx xx1x Not applicable otherwise
PERSIST	1 or 0	0, 1; if Ityp contains xxxx xxxx xxxx 1xxx Not applicable otherwise
LIFE	32 or 0	0-(2^{31}–1); if Ityp contains xxxx xxxx xxxx 1xxx Not applicable otherwise
NEXT-USE	32 or 0	0-(2^{31}–1); if Ityp contains xxxx xxxx xxxx 1xxx Not applicable otherwise
XC	32 or 0	0-(2^{32}–1); if Ityp contains xxxx xxxx xx1x xxxx Not applicable otherwise
YC	32 or 0	0-(2^{32}–1); if Ityp contains xxxx xxxx xx1x xxxx Not applicable otherwise
WC	32 or 0	0-(2^{32}–1); if Ityp contains xxxx xxxx xx1x xxxx Not applicable otherwise
HC	32 or 0	0-(2^{32}–1); if Ityp contains xxxx xxxx xx1x xxxx Not applicable otherwise
ROT	32 or 0	0-31, see Table A.6; if ITyp contains xxxx xxxx x1xx xxxx. Not applicable otherwise

Table A.6 — Encoding of the ROT field

Binary value	Meaning
0000 0000 0000 0000	Orientation not specified
0000 0000 000x 0001	Rotate by 0° clockwise
0000 0000 000x 0010	Rotate by 90° clockwise
0000 0000 000x 0011	Rotate by 180° clockwise
0000 0000 000x 0100	Rotate by 270° clockwise
0000 0000 0001 xxxx	Flip image left to right after rotation
	All other values are reserved for future use by ISO/IEC.

A.2.5 Object Metadata box

The Object Metadata box signals information about the media contents composing the JPEG Snack representation.

The type of the Object Metadata box shall be ‘obmb' (0x6f62 6d62). The contents of the box shall be as in Figure A.5, the fields are summarized in Table A.8.

Figure A.5 — Organization of the contents of the Object Metadata box

— TOGGLES (T): This field shall contain TOGGLES as in Table A.7.

Table A.7 — TOGGLES

Binary value	Meaning	TOGGLE Details
0000 0xx1	Number of media present	Number of media present. This option signals if the NUMBER OF MEDIA field is present. No number of media present implies that the number of media is 1.
0000 0xx0	No number of media present	
0000 0x1x	Style present	Style present. This option signals if the STYLE field is present.
0000 0x1x	No style present	
0000 01xx	Opacity present	Opacity present. This option signals if the OPACITY field is present.
0000 00xx	No opacity present	
The upper 5 bits are reserved for future use by ISO/IEC.		

— Other fields: Details of fields are described in Table 5.1.

Table A.8 — Format of the contents of the Object Metadata box

Field name	Size (bits)	Value
Toggle	8	See Table A.7
ID	8	See Table 5.1
Media type	Variable	
Number of media	8 or 0	
Opacity	32 or 0	
Style	Variable or 0	
Location	Variable	

Annex B
(informative)

Container of JPEG Snack

B.1 Overview

This annex provides a description for the organization of the JPEG Snack files.

As described in 6.1.1, the JPEG Snack format is composed based on the default image. Therefore, JPEG Snack formats are embedded into the default image format.

B.2 JPEG Snack files

B.2.1 ISO/IEC 10918-1

— Media type: `image/jpeg`

— File organization: JUMBF boxes of the JPEG Snack format are embedded in APP11 Segments, as illustrated in Figure B.1. The syntax of the APP11 marker segment is defined in ISO/IEC 18477-3.

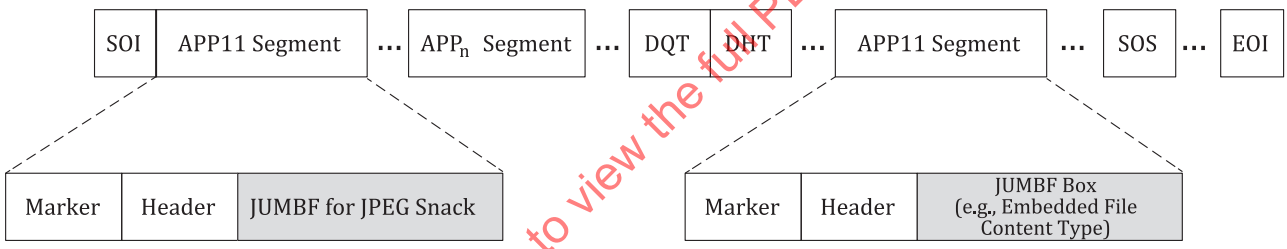


Figure B.1 — Overview of the JPEG Snack file organization with an ISO/IEC 10918-1

In Figure B.1, a box for the JPEG Snack format is placed at the beginning of the file, except the SOI marker. This organization allows the JPEG Snack application to parse necessary information as quickly as possible while a legacy ISO/IEC 10918-1 decoder ignores all APP11 markers in the file. Furthermore, all APP11 marker segments shall be positioned before the image data.

B.2.2 Box-based formats

— Media type: A media type corresponding to the default image.

— File organization: JUMBF boxes of the JPEG Snack format are embedded as illustrated in Figure B.2.

Figure B.2 — Overview of the JPEG Snack file organization with a box-based format.

When the format of the default image is based on the box-based format, the JPEG Snack Content type box shall be at first except Signature box and File Type box. Other boxes related to the JPEG Snack content type box may be placed at any position in the file.

Annex C
(informative)

Usage examples

C.1 Overview

This annex provides a description for practical examples of the JPEG Snack formats.

Examples are as follows:

— A simple slide show (see C.2.1)

— Rendering multiple objects simultaneously (see C.2.2)

— A moving object (see C.2.3)

Required values for the boxes are provided in Tables C.1 to C.11.

C.2 Examples

C.2.1 A simple slide show

This example explains how to specify a simple slide show using consecutive images: an image sequence. The show starts with the default image being rendered to the display. It consists of 7 images, including the default image, and lasts for 14 s while each image is displayed for 2 s at a fixed position.

When the value of the REPT parameter in the Instruction Set box is 65535, the instruction is repeated indefinitely.

The organization of the boxes is illustrated in Figure C.1.