**ISO/IEC JTC 1/SC 29/WG3 N1075**

Date : 2023-11-03

**ISO/IEC 23001-17:2022/AMD 2:2023(E)**

ISO/IEC JTC1/SC 29

Secretariat: JISC

**Information technology — MPEG Systems technologies — Part 17: Carriage of uncompressed video and images in ISO Base Media File Format — Amendment 2: Generic Sample Compression**

WD stage

**Copyright notice**

This ISO document is a working draft or committee draft and is copyright-protected by ISO. While the reproduction of working drafts or committee drafts in any form for use by participants in the ISO standards development process is permitted without prior permission from ISO, neither this document nor any extract from it may be reproduced, stored or transmitted in any form for any other purpose without prior written permission from ISO.

Requests for permission to reproduce this document for the purpose of selling it should be addressed as shown below or to ISO's member body in the country of the requester:

ISO copyright office

Case postale 56 • CH-1211 Geneva 20

Tel. + 41 22 749 01 11

Fax + 41 22 749 09 47

E-mail copyright@iso.org

Web www.iso.org

Reproduction for sales purposes may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

Editor’s introduction – Generic Sample Compression

With 23001-17 (Carriage of uncompressed video and images in ISO Base Media File Format) nearing completion, we are looking at potential improvements beyond simple uncompressed imagery. While 23001-17 nominally defines the mechanism to carry uncompressed imagery and raster data within ISOBMFF, it also more generically defines a mechanism to define the in-memory layout for an image item or sample, and then store the image using that layout:

* Component value types and sizes
* Tiling to enable efficient spatial-based access for large imagery
* Padding to ensure that individual component values or groups of component values can be accessed by the processing unit without having to perform bit shifts for every single access, and without having to cross key storage/memory page boundaries

This capability brings ISOBMFF on par with other generic storage formats for numeric data (such as HDF5) with one exception – data-agnostic numerically-lossless and bitwise-lossless compression, transparent to the end user. In HDF5, for example, a large N-dimensional array of numerical data can be chunked (i.e., tiled) and then each chunk is compressed using off-the-shelf ubiquitous data-agnostic compression tools (e.g., deflate). This provides storage and transmission savings similar to numerically lossless image coders, with minimal computational performance impact. These capabilities can be applied not only to typical integer-based pixel formats, but also to IEEE 754 floating-point pixel formats that are unsupported in most imagery compression algorithms.

Adding this capability to 23001-17 is expected to provide cost savings (for both storage and network transmission), particularly for applications and datasets involving large amounts of uncompressed content, such as geospatial and scientific imagery, without significantly changing how those applications access the pixel data.

While not necessarily required, a mechanism that could be applied to other forms of data (e.g., audio or KLV formatted metadata), this capability is not necessarily required.

**Use cases**

* Data producer generates a large image (or image sequence) using 32-bit IEEE 754 binary floating point component values after calibration. The image is tiled using 1024 × 1024 tiles and each tile is independently compressed using deflate.
* Data consumer desires to load only a specific spatial region from an image or sample based on some form of chunking of the image (chunk by rectangular tiles or rows). Consumer uses offset/size information provided within the ISOBMFF file to locate only the desired chunks. Each of those chunks can be independently decompressed.
* Data consumer desires to load only a spatial region from a large tiled image, where the desired region is smaller than a region contained within a single compressed chunk. After decompressing the chunk, the order, alignment and padding of the component data is maintained, enabling the consumer to calculate individual component value offsets – parsing through the decompressed chunk to locate specific pixels/component is not necessary.
* A data producer collects an image of the Earth scanning diagonally from northwest to southeast. For simpler human viewing of the image, the collected image is rotated 45 degrees to align north to up, and additional padding is added to form a tiled, rectangular image buffer. Since the four corner tiles are entirely fill data, they are omitted from the data stored within the ISOBMFF file.
* Legal requirements for records management require bitwise-lossless compression of the image data – it is not sufficient that the decompressed pixel value is numerically equal to the original pixel value; the specific bit patterns must also be the same.

**Requirements**

* Numerically and bitwise lossless compression of raw image items and track samples, especially when consisting of floating-point formatted media.
* Pixel organization prior to compression (as well as post-decompression) mirrors the uncompressed spec in 23001-17. The point is for the encoder to determine how to best organize the pixel/component data when in a directly-accessible form, and then to provide simple off-the-shelf, numerically/bitwise-lossless compression for that data.
* Utilization of existing compression technology with open licensing, broad/mature support, and availability of open-source software and tools
* Ability to access portions of the image without fully decompressing the entire sample or image item.
  + Ability to code and access tiles independently. This includes gridded items as well as tiles defined within 23001-17.
  + Investigate the ability for “unnecessary” tiles to be omitted from the data
    - e.g., a large geospatial image is rotated within the rectangular pixel boundaries so north is up. This causes the corners of the expanded rectangular image to be just fill pixels, with a resulting preference to not have to store those fill pixels
    - Specifying a transformative property for arbitrary rotation is ultimately not sufficient as there are many means to precisely georeference an image (e.g., orthorectification) involving pushing pixels around based on imaging geometry and terrain.
    - Determine if sub images of a gridded image item can be omitted.
* Orthogonal capability to the 23001-17 component value alignment, padding and component value organization is highly desired. Upon decompression, each chunk is used exactly as if the respective uncompressed values had been loaded directly from storage or the network.
* Constructive interaction between transformational properties is desired. For example, if I compress the sample as individual chunks, but then encrypt the entire sample, the independence of those chunks might be lost. We note that this might not be generically possible without defining tiling-aware encryption/protection mechanism, and that might look a lot like a gridded image item.

Contents

[Foreword vi](#_Toc149565120)

[3.12 Agnostically-compressed item 1](#_Toc149565121)

[3.13 Agnostically-compressed sample 1](#_Toc149565122)

[8.1 Overview 1](#_Toc149565123)

[8.2 Compression Configuration Box 2](#_Toc149565124)

[8.2.1 Definition 2](#_Toc149565125)

[8.2.2 Syntax 2](#_Toc149565126)

[8.2.3 Semantics 2](#_Toc149565127)

[8.3 Agnostically-compressed media tracks 3](#_Toc149565128)

[8.3.1 Overview 3](#_Toc149565129)

[8.3.2 Compressed Byte Ranges Information Box 3](#_Toc149565130)

[8.3.2.1 Definition 3](#_Toc149565131)

[8.3.2.2 Syntax 4](#_Toc149565132)

[8.3.2.3 Semantics 5](#_Toc149565133)

[8.4 Agnostically-compressed items 5](#_Toc149565134)

[8.4.1 Overview 5](#_Toc149565135)

[8.4.2 Item Compressed Byte Ranges 5](#_Toc149565136)

[8.4.2.1 Definition 5](#_Toc149565137)

[8.4.2.2 Syntax 6](#_Toc149565138)

[8.4.2.3 Semantics 6](#_Toc149565139)

[9.1 Mapping to Common Encryption 6](#_Toc149565140)

[9.2 Mapping to Generic Sample Compression 6](#_Toc149565141)

[9.3 Combined usage of Generic Sample Compression and Common Encryption 7](#_Toc149565142)

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 documents should be noted.  This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see [www.iso.org/directives](http://www.iso.org/directives) or [www.iec.ch/members\_experts/refdocs](http://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](http://www.iso.org/patents)) or the IEC list of patent declarations received (see [patents.iec.ch](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 the following URL:  [www.iso.org/iso/foreword.html](http://www.iso.org/iso/foreword.html). In the IEC, see [www.iec.ch/understanding-standards](https://www.iec.ch/understanding-standards)

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

A list of all parts in the ISO/IEC 23001 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](https://www.iso.org/members.html) and [www.iec.ch/national-committees](http://www.iec.ch/national-committees).

Information technology — MPEG Systems technologies — Part 17: Carriage of uncompressed video and images in ISO Base Media File Format — Amendment 2: Generic Sample Compression

*Add the following entries into clause 3 (Normative Reference)*

* *Deflate compression IETF RFC 1951*
* *Deflate compression and zlib bitstream format IETF RFC 1950*
* *Brotli compression IETF RFC 7932*

*Add the following new subclauses after subclause 3.11*

## Generically-compressed

compressed using an off-the-shelf compression capability that can be applied to any source data and is not necessarily optimized for specific types of data or information

## Generically-compressed item

Item that has been generically-compressed

## Generically-compressed sample

sample that has been generically-compressed

*Add following new clause after clause 7*

# Generic compression of items and sample data

## Overview

While storing fully uncompressed item and sample data is useful in many scenarios that are not well-supported by typical image compression algorithms (e.g., floating point or very high-bit depth imagery), the ability to compress that data without any compression loss is often still desired to reduce storage sizes and transmission times, such as in the following scenarios:

* Data producer generates a large image (or image sequence) using 32-bit IEEE 754 binary floating point component values after calibration. The image is tiled using 1024 × 1024 tiles and each tile is independently compressed using deflate.
* Data consumer desires to load only a specific spatial region from an image or sample based on some form of chunking of the image (chunk by rectangular tiles or rows). Consumer uses offset/size information provided within the ISOBMFF file to locate only the desired chunks. Each of those chunks can be independently decompressed.
* Data consumer desires to load only a spatial region from a large tiled image, where the desired region is smaller than a region contained within a single compressed chunk. After decompressing the chunk, the order, alignment and padding of the component data is maintained, enabling the consumer to calculate individual component value offsets – parsing through the decompressed chunk to locate specific pixels/component is not necessary.
* A data producer collects an image of the Earth scanning diagonally from northwest to southeast. For simpler human viewing of the image, the collected image is rotated 45 degrees to align north to up, and additional padding is added to form a tiled, rectangular image buffer. Since the four corner tiles are entirely fill data, they can more efficiently be stored than simply compressing multiple blocks of fill data.

To decompress a complete media sample or item, the file reader locates the data as given by the sample or item’s compressed byte ranges description. The extracted compressed data is then decompressed according to the compression algorithm specified by the compression\_type field in the CompressionConfigBox. The resultant data is exactly the data as was specified by that element of the sample or image as specified by the underlying media format, including any padding placed at the end of that element to align the next element.

Specifics on the uncompressed bytes associated with each byte ranges are specified by the subsample\_type field in the CompressionConfigBox.

### Compression Configuration Box

### Definition

Box Type: 'cmpC'  
Container: SchemeInformationBox or ItemPropertyContainerBox   
Mandatory: Yes (when the SchemeType is 'gcmp')  
Quantity: One

The CompressionConfigurationBox specifying the specific data compression method used and codec-specific type of compressed byte ranges within a media sample or item data.

The definition of each value of compression\_type specifies not only the algorithm but also the bitstream format for each compressed subsample. For example, ‘zlib’ specifies the use of the deflate algorithm as packaged in the zlib format defined by IETF RFC 1950.

Value 0 for compressed\_range\_type indicates that the compressed range is always the complete media sample or item.

Derived specifications may assign compressed\_range\_type values other than 0 according to the specificities of the underlying media format.

If compressed\_range\_type is 0, then can\_decompress\_contiguous\_ranges shall be 1.

EDITOR’S NOTE: If this capability can be applied to other types of data, consider making the values dependent on the type of item (e.g., the above values for video samples and image items, but for KLV tracks the values might be keys).

### Syntax

aligned(8) class CompressionConfigurationBox extends FullBox('cmpC') {  
  
unsigned int(32) compression\_type;  
unsigned int(1) can\_decompress\_contiguous\_ranges;  
unsigned int(7) compressed\_range\_type;  
}

Ed question: Consider swapping the order of can\_decompression\_contiguous\_ranges and compressed\_range\_type. Consider renaming “range” to “entity” – I don’t think we intend the ability to compress arbitrary byte-ranges of the original data, but only ranges tied to described entities within the sample or item (e.g., tiles, rows, keys)

### Semantics

compression\_type is a 4CC indicating the compression mode for the sample or item. The following values are defined:

• 'defl': deflate algorithm as defined in IETF RFC 1951

• 'zlib': deflate algorithm as packaged in the format defined by IETF RFC 1950.

• 'brot': Brotli algorithm as defined in IETF RFC 7932

can\_decompress\_contiguous\_ranges specifies whether contiguous compressed byte ranges in the media sample or item data can be decompressed as a single large input buffer (value 1), or whether each compressed byte range must be decompressed individually (value 0). If the value is 1, the decompressed data for the entire sample or item shall be equal to the concatenation of each decompressed byte range.

compressed\_range\_type indicates the entity being compressed within a generically-compressed media sample or item. Each range contains exactly the result of the compression algorithm applied to the identified entity. Legal values of the compressed\_range\_type are specified in Clause 9.2

### Generically-compressed media tracks

### Overview

Generically-compressed media tracks compliant to this specification are media tracks compliant to ISO/IEC 14496-12 that use a reserved transformation scheme of type 'gcmp', hereafter called generically-compressed sample entry.

The SchemeInformationBox of a generically-compressed sample entry shall contain one CompressionConfigurationBox, specifying the specific data compression method used and the granularity of the compression scheme over the sample data.

The CompressedByteRangesInformationBox is used to specify the location and size of each independently compressed element within the sample.

For non-fragmented movies, the sample table of a track using a generically-compressed sample entry shall contain one CompressedByteRangesInformationBox. For a fragmented movie, each TrackFragmentBox of a track using a generically-compressed sample entry shall contain one CompressedRangesInformationBox.

A generically-compressed sample entry does not impact the description of the media samples of the track: random access information, subsample description and sample group properties describe the media sample without the generic compression being applied.

EDITOR’S NOTE: If the ability to switch compression type across samples for a same media configuration is desired, it would be better to use an essential sample group. Contributions on this topic are expected.

### Compressed Byte Ranges Information Box

### Definition

Box Type: 'cbri'  
Container: SampleTableBox or TrackFragmentBox  
Mandatory: See below  
Quantity: Zero or One

The CompressedByteRangesInformationBox describes the byte ranges of compressed data for samples of a track or track fragment.

The CompressedByteRangesInformationBox shall be present whenever the track has a sample entry using a restricted scheme of type 'gcmp'and the compressed\_range\_type of the associated CompressionConfigurationBox has a value other than 0.

The CompressedByteRangesInformationBox shall not describe more samples than present in the track or track fragment.

If the CompressedByteRangesInformationBox describes less samples than available in the track or track fragment, this implies that the remaining non described samples have a single range covering the entire sample.

For each described sample, the sum of the size of all ranges shall be equal to the size of the sample, as indicated in the SampleTableBox or in the TrackRunBox.

EDITOR’S NOTE: the use of offsets within the range description changes the behaviour of a file reader, as the bytes cannot be fetched from mdat without looking at the CompressedByteRangesInformationBox . For safety purposes, we should either have 2 different restricted scheme types, or at least have a signaling in the CompressionConfigurationBox . Contributions on this topic are expected.

### Syntax

aligned class compressed\_range\_info(offsetsize\_nbbits, rangesize\_nbbits) {  
  
 if (offsetsize\_nbbits)  
 signed int(offsetsize\_nbbits) range\_offset;

unsigned int(rangesize\_nbbits) range\_size;  
}

class CompressedByteRangesInformationBox extends FullBox('cbri', version=0, flags=0) {

unsigned int(2) offsetsize\_code;

unsigned int(2) rangesize\_code;

bit(4) reserved = 0;

unsigned int(32) num\_entries;

int offsetsize\_nbbits;

if (offsetsize\_code==0) offsetsize\_nbbits=0;

else if (offsetsize\_code==1) offsetsize\_nbbits=16;

else if (offsetsize\_code==2) offsetsize\_nbbits=32;

else offsetsize\_nbbits=64;

int rangesize\_nbbits;

if (rangesize\_code==0) rangesize\_nbbits=8;

else if (rangesize\_code==1) rangesize\_nbbits=16;

else if (rangesize\_code==2) rangesize\_nbbits=32;

else rangesize\_nbbits=64;

for (int i=0; i<num\_entries; i++) {

unsigned int(32) num\_samples;

unsigned int(32) num\_ranges;

for (int s = 0; s < num\_samples; i++) {

for (int r = 0; r < num\_ranges; r++) {

compressed\_range\_info(offsetsize\_nbbits, rangesize\_nbbits);

}

}

}

}

### Semantics

offsetsize\_code indicates the number of bits used to describe range offsets. A value of 0 implies that offsets are not used.

rangesize\_code indicates the number of bits used to describe range sizes.

num\_entries is an integer that indicates the number of entries described in the box. Each entry describe samples which share the same number of subsamples.

num\_samples is an integer that indicates the number of samples for the current entry. This value shall not be 0.

num\_ranges is an integer that indicates the number of compressed byte ranges for samples described by the current entry. If this value is 0, this implies that a single byte range covering the entire sample data is used.

NOTE: If the structure of the sample varies within the track or track segment (e.g., the dimensions of the sample changes, or the number of keys within KLV samples changes), then the number of compressed ranges might also change dependent on how elements within the sample are mapped to ranges.

range\_offset is the offset in bytes to the first byte of the compressed byte range. When the sample is present in a track fragment, this offset is relative to the start of the parent MovieFragmentBox. Otherwise, this offset is relative to the start of the file.

range\_size is size in bytes of this compressed byte range.

### Generically-compressed items

### Overview

A generically-compressed item compliant to this specification is an item with an associated essential property of type 'cmpC'. When content protection is applied to a generically-compressed item, the protection shall be applied to the compressed payload.

EDITOR’S NOTE: the current design uses an essential property to identify the compression. It could be possible to define a non-displayable item and have a derived image on this item. Contributions on this topic are welcome.

### Item Compressed Byte Ranges

### Definition

Box Type: 'icbr'  
Container: ItemPropertyContainerBox   
Mandatory: No  
Quantity: Zero or One

The ItemCompressedByteRangesBox describes the byte ranges of compressed data for an item.

The ItemCompressedByteRangesBox shall be present for any item with an associated essential property of type 'cmpC' (CompressionConfigurationBox) for which compressed\_range\_type has a value other than 0. When not present, this implies that a single byte range covering the complete item data is used.

The sum of the size of all ranges described by a ItemCompressedByteRangesBox shall be equal to the size of the item.

EDITOR’S NOTE: the use of offsets within the range description changes the behavior of a file reader, as the bytes for an item cannot be fetched from mdat without looking at the ItemCompressedByteRangesBox. For safety purposes, we should have a signaling in the CompressionConfigurationBox . Contributions on this topic are expected.

### Syntax

aligned(8) class ItemCompressedByteRangeInfo extends FullBox('icbr', version, flags=0) {

unsigned int(32) num\_ranges;

for (int r = 0; r < num\_ranges; r++) {  
 if (version==1) {

compressed\_range\_info(64, 64);

} else if (version==0) {  
 compressed\_range\_info(32, 32);

}

}  
}

### Semantics

num\_ranges is an integer that indicates the number of compressed byte ranges for the item. If this value is 0, this implies that a single byte range covering the entire item data is used.

range\_offset is the offset in bytes to the first byte of the compressed byte range. This offset is relative to the start of the file.

range\_size is size in bytes of this compressed byte range.

# Data byte ranges for uncompressed video and image items

## Mapping to Common Encryption

When a media sample or image item is encrypted using ISO/IEC 23001-7 Common Encryption, the encryption boundaries, i.e. the start of a CENC subsample, should be aligned with significant semantic elements of the item, e.g. start of a tile, a row, a component plane, etc. Given the uncompressed nature of the media, it is expected that most CENC subsamples will have a BytesOfClearData with value 0. Derived specifications may further restrict the mapping of these elements to CENC subsamples.

## Mapping to Generic Sample Compression

For uncompressed video sample entries or image items compliant to this document, the following compressed\_range\_type values for the CompressionConfigurationBox are defined:

* 1: the entity is the full image for a given component (component-based interleave and mixed interleave)
* 2: the entity is a tile
* 3: the entity is a row
* 4: the entity is a pixel
* 5: the entity is a single KLV-encoded key
* other values are reserved.

## Combined usage of Generic Sample Compression and Common Encryption

When a media sample or image item is using both generic sample compression and encryption using ISO/IEC 23001-7 Common Encryption, the following apply:

* The encryption shall be applied to the generic compressed data,
* The first byte of each compressed byte range shall be the first byte of a CENC subsample.