**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 3**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC 1/SC 29/WG 3 m** **67070**

**Rennes – April 2024**

**Title: Draft for 23001-17 Amd2 DAM**

**Author: Scott Houchin**

**scott.houchin@aero.org**

**+1 571-304-3914**

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

Date : 2024-03-22

**ISO/IEC 23001-17:2022/AMD 2:2024(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 compression of samples and items in ISOBMFF**

DAM 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.

Introduction

This amendment to ISO/IEC 23001-17 includes the following additions and corrections to the standard.

* Defines the capability to apply generic compression to still and motion imagery as well as other media types (e.g., KLV metadata tracks)
* Clarifies the original standard text related to padding of images using subsampled chroma components
* Adds the capability to encode signed integer components.
* Adds brands for both uncompressed imagery itself and for generically compressed media

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.

In this amendment, this mechanism is applied to KLV formatted metadata tracks in addition to image samples and image items. Application to other media types can be defined in future standards.

**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 chunk by 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 tile. After decompressing the tile, the order, alignment and padding of the component data is maintained, enabling the consumer to calculate individual component value offsets – parsing through the decompressed tile 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 image items and track samples, especially when consisting of floating-point formatted media.
* Pixel organization prior to compression (as well as post-decompression) is defined by 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 implement simple off-the-shelf, numerically/bitwise-lossless compression on 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 compress, access, and decompress tiles independently. This includes gridded items as well as tiles defined within 23001-17.
  + Ability to minimize coding/inclusion of fill 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 the sample is compressed as individual chunks, but the compressed sample is then is encrypted as a single chunk, 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.
* Ability to compress KLV metadata tracks and items.

Contents

[3.12 Generically-compressed 1](#_Toc163730376)

[3.13 Generically-compressed item 1](#_Toc163730377)

[3.14 Generically-compressed sample 1](#_Toc163730378)

[4.4 Compatible brands 1](#_Toc163730379)

[8.1 Overview 4](#_Toc163730380)

[8.2 Compression Configuration Box 5](#_Toc163730381)

[8.2.1 Definition 5](#_Toc163730382)

[8.2.2 Syntax 5](#_Toc163730383)

[8.2.3 Semantics 5](#_Toc163730385)

[8.3 Generically-compressed media tracks 6](#_Toc163730386)

[8.3.1 Overview 6](#_Toc163730387)

[8.3.2 Compressed Byte Ranges Information Box 6](#_Toc163730388)

[8.4 Generically-compressed items 8](#_Toc163730389)

[8.4.1 Overview 8](#_Toc163730390)

[8.4.2 Item Compressed Byte Ranges 9](#_Toc163730391)

[9.1 Mapping to Common Encryption 10](#_Toc163730392)

[9.2 Mapping to Generic Sample Compression 10](#_Toc163730393)

[9.3 Combined usage of Generic Sample Compression and Common Encryption 10](#_Toc163730394)

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 compression of samples and items in ISOBMFF

*2*

*Add the following new subclauses after subclause 3.11*

## Generically-compressed

compressed using one of a defined set of off-the-shelf numerically and bitwise lossless compression capabilities

## Generically-compressed item

item that has been numerically and bitwise losslessly compressed

## Generically-compressed sample

sample that has been numerically and bitwise losslessly compressed

*Add new clause 4.4 specifying compatible brands for uncompressed and generically compressed data:*

# Uncompressed video and image formats

## Compatible brands

TBD

*In clause 5.2.1.3, Table 2, add the following row and modify the last row to reflect use of the previously reserved value 3:*

**Table 2 - Component formats**

|  |  |
| --- | --- |
| Value | Description |
| 3 | Component value is a signed two’s complement integer coded on component\_bit\_depth bits. |
| other values | ISO/IEC reserved for future definition |

*In clause 5.2.1.5.3,* *replace the following lines:*

If row\_align\_size is not 0 and interleave\_type is 0:

* row\_align\_size shall be a multiple of 2
* the row alignment for components of type ‘U’ and ‘V’, as defined in 5.2.1.7, shall be done using row\_align\_size/2

If tile\_align\_size is not 0:

* tile\_align\_size shall be a multiple of 2
* the tile alignment for components of type ‘U’ and ‘V’, as defined in 5.2.1.7, shall be done using tile\_align\_size/2

*with the following (the second bullet in each list is deleted):*

If row\_align\_size is not 0 and interleave\_type is 0:

* row\_align\_size shall be a multiple of 2

If tile\_align\_size is not 0:

* tile\_align\_size shall be a multiple of 2

*In clause 5.2.1.5.4,* *replace the following lines:*

If row\_align\_size is not 0 and interleave\_type is 0:

* row\_align\_size shall be a multiple of 2
* the row alignment for components of type ‘U’ and ‘V’, as defined in 5.2.1.7, shall be done using row\_align\_size/2

If tile\_align\_size is not 0:

* tile\_align\_size shall be a multiple of 4
* the tile alignment for components of type ‘U’ and ‘V’, as defined in 5.2.1.7 shall be done using tile\_align\_size/4

*with the following (the second bullet in each list is deleted):*

If row\_align\_size is not 0 and interleave\_type is 0:

* row\_align\_size shall be a multiple of 2

If tile\_align\_size is not 0:

* tile\_align\_size shall be a multiple of 4

*In clause 5.2.1.5.5, replace the following lines:*

If row\_align\_size is not 0 and interleave\_type is 0:

* row\_align\_size shall be a multiple of 4
* the row alignment for components of type ‘U’ and ‘V’, as defined in 5.2.1.7, shall be done using row\_align\_size/4

If tile\_align\_size is not 0:

* tile\_align\_size shall be a multiple of 4
* the tile alignment for components of type ‘U’ and ‘V’, as defined in 5.2.1.7, shall be done using tile\_align\_size/4

*with the following (the second bullet in each list is deleted):*

If row\_align\_size is not 0 and interleave\_type is 0:

* row\_align\_size shall be a multiple of 4

If tile\_align\_size is not 0:

* tile\_align\_size shall be a multiple of 4

*In clause 5.2.1.7, 2nd paragraph after NOTE4, replace:*

Rows of tiles shall be byte-aligned at the end of the row:

*with the following:*

Rows and tile rows shall be byte-aligned at the end of the row:

*In clause 5.2.1.7, replace:*

If row\_align\_size is 0, no additional padding is present at the end of rows of tiles. Otherwise, let RowSize be the number of bytes required to contain, for a given row R:

* all values of all components of row R if interleave\_type is 1 or 5 or if interleave\_type is 2 and component type is ‘U’ or ‘V’ (including all component, block and pixel padding within and at the end of the sample data for row R),
* all values of the current component for row R (including all component and block padding within the sample data for row R) otherwise.

*with the following:*

If row\_align\_size is 0, no additional padding is present at the end of rows and tile rows. Otherwise, let RowSize be the number of bytes required to contain, for a given row R:

* all values of all components of row R (including all component, block and pixel padding within and at the end of the sample data for row R) if interleave\_type is 1 or 5
* all values of pixel interleaved components (‘U’ and ‘V’) of row R if interleave\_type is 2 and component type is ‘U’ or ‘V’ (including all component, block and pixel padding within and at the end of the sample data for row R),
* all values of the current component for row R (including all component and block padding within the sample data for row R) otherwise.

*Add following new clause after clause 7*

# Generic compression of items and sample data

## Overview

Storing uncompressed item and sample data is required in many use cases, some of which are not well-supported by typical image compression algorithms (e.g., floating point or very high-bit depth imagery). Other types of sample data, such as KLV, are only compressible by standard generic data compression mechanisms. The ability to compress data for any media type without compression loss is desirable for reducing storage sizes and transmission times, such as in the following scenarios:

* A 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.
* A 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 chunk by rows). Consumer uses offset/size information provided within the file to locate only the desired chunk(s). Each chunk is independently decompressed.
* A 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 locate individual component values via calculated offsets.
* 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.
* TBD KLV example

To decompress a complete media sample or item, the file reader locates the data as given by the sample or item’s compressed extents 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 formatted exactly as was specified by the underlying media format, including any padding placed at the end of the element to align the next element.

If the value of the compressed\_entity\_type field in the CompressionConfigBox is not 0, decompression of individual portions of the sample is possible. In those cases, the individual ranges specified in the GenericallyCompressedExtentsInfoBox map to the individual entities as specified by the compressed\_entity\_type field, in the order those entities would have been found in the coded data were that coded data left uncompressed as specified in Clause 5.

## Compression Configuration Box

### Definition

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

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

This box can be:

* added to a video sample entry for media tracks
* added as properties associated with an image item.

The syntax in this section are given for a video sample entry container and the defined boxes therefore extend Box (resp. FullBox). When used in an ItemPropertyContainerBox, the same syntax applies but the defined boxes extend ItemProperty (resp. ItemFullProperty). The properties defined in the following sections are descriptive properties.

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\_entity\_type indicates that the compressed range is always the complete media sample or item.

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

If compressed\_entity\_type is 0, then must\_decompress\_individual\_entities shall be 0.

### Syntax

aligned(8) class CompressionConfigurationBox extends FullBox('cmpC', 0, 0) {  
  
unsigned int(32) compression\_type;  
unsigned int(1) must\_decompress\_individual\_entities ;  
unsigned int(7) compressed\_entity\_type;  
}

### Semantics

compression\_type is a 4CC indicating the compression mode for the sample or item. Values are defined in Table 6.

**Table 6 – Compression types**

| Value | Description |
| --- | --- |
| ‘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 |

compressed\_entity\_type indicates the entity being compressed within a generically-compressed media sample or item. The range of bytes specified for each entity contains exactly the result of the compression algorithm applied to the identified entity. Legal values of the compressed\_entity\_type are specified in Clause 9.2. For example, if the value of this field is 2 (entities are tiles), then each tile of the sample or image item is independently accessible and decompressible.

must\_decompress\_individual\_entities specifies whether each individual entity must be decompressed individually.

If the value is 1, then the reader must extract the bytes for each individual entity from the file and perform the decompression operation (as specified by the compression\_type field) on the bytes for that individual entity. For example, if entities are tiles, and the reader wants to decompress tiles 2,3 and 4, then the reader must load the bytes for entity 2 and decompress just those bytes, resulting in the uncompressed bytes for tile 2. The reader then repeats that process for tile 3 and tile 4.

If the value is 0, then the reader can concatenate, in entity order, the compressed bytes for any contiguous list of entities (in entity order), and decompress those entities using a single decompression option. For the example of a reader that wants to decompress tiles 2, 3 and 4, the reader loads the bytes for those three tiles (regardless of what order those bytes are stored) and concatenate those three sets of bytes in the order 2, 3, 4. The reader can then decompress those three tiles using a single decompression operation, resulting in a an uncompressed set of bytes containing uncompressed tile 2, uncompressed tile 3, and uncompressed tile 4, in that order. Using the information provided in the UncompressedFrameConfigBox, the reader can calculate the size in bytes of each tile and then extract the individual tiles from the decompressed bytes.

NOTE: The ability for the reader to decompress multiple entities as a single combined input buffer is largely dependent on the compression algorithm and codestream format. For example, while the IETF RFC 1950 format for DEFLATE does include a short header at the start of each compressed entity, the decompressor does recognize that header even if it is in the middle of the compressed data buffer. In many cases, it may be possible for entities to be concatenated in any order as long as the reader knows what order the entities were found in the compressed buffer, and can then extract those entities from the decompressed buffer. However, this capability is not guaranteed by this standard.

## 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 data compression method used and the partitioning of the compression scheme over the sample data.

The GenericallyCompressedExtentsInfoBox 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 GenericallyCompressedExtentsInfoBox . For a fragmented movie, each TrackFragmentBox of a track using a generically-compressed sample entry shall contain one GenericallyCompressedExtentsInfoBox.

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.

### Generically Compressed Extents Info Box

#### Definition

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

The GenericallyCompressedExtentsInfoBox describes the extents of compressed data for samples of a track or track fragment.

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

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

If the GenericallyCompressedExtentsInfoBox describes less samples than available in the track or track fragment, this implies 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 GenericallyCompressedExtentsInfoBox . 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\_gc\_extent\_info(extent\_offset\_nbbits, extent\_size\_nbbits) {  
  
 if (extent\_offset\_nbbits)  
 signed int(extent\_offset\_nbbits) extent\_offset;

unsigned int(extent\_size\_nbbits) extent\_size;  
}

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

unsigned int(2) extentoffsetsize\_code;

unsigned int(2) extentsize\_code;

bit(4) reserved = 0;

unsigned int(32) num\_entries;

int extentoffsetsize\_nbbits;

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

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

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

else extentoffsetsize\_nbbits=64;

int extentsize\_nbbits;

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

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

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

else extentsize\_nbbits=64;

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

unsigned int(32) num\_samples;

unsigned int(32) num\_gc\_extents;

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

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

compressed\_gc\_extent\_info(

extentoffsetsize\_nbbits,

extentsize\_nbbits

);

}

}

}

}

#### Semantics

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

extentsize\_code indicates the number of bits used to describe extent sizes.

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

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

num\_gc\_extents is an integer indicating the number of compressed extents for samples described by the current entry. If this value is 0, this implies that a single extent 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 or size of the sample changes), then the number of compressed extents might also change depending on how elements within the sample are mapped to extents.

compressed\_gc\_extent\_info.extent\_offset is the offset in bytes to the first byte of the compressed extent. 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.

compressed\_gc\_extent\_info.extent\_size is size in bytes of this compressed extent.

## Generically-compressed items

### Overview

Generically-compressed items have an associated essential item 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.

### Generically Compressed Item Extents

#### Definition

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

The GenericallyCompressedItemExtentsInfoBox describes the extents of compressed data for an item.

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

The sum of the size of all extents described by a GenericallyCompressedItemExtentsInfoBox shall be equal to the size of the item.

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

#### Syntax

The dependent class compressed\_gc\_extent\_info is as defined in clause 8.3.2.

aligned(8) class GenericallyCompressedItemExtentsInfoBox extends ItemFullProperty('icbr', version, flags=0) {

unsigned int(32) num\_gc\_extents;

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

compressed\_gc\_extent\_info(64, 64);

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

}

}  
}

#### Semantics

num\_gc\_extents is an integer that indicates the number of compressed extents for the item. If this value is 0, this implies that a single extent covering the entire item data is used.

compressed\_gc\_extent\_info.extent\_offset is the offset in bytes to the first byte of the compressed extent. The origin of this offset is dependent on the source of the data. For example, the offset can be from the start of the file, or from the start of the data[] in an ItemDataBox. In all cases, it matches the data origin as defined in ISO/IEC 14496-12 for ItemLocationBox.

compressed\_gc\_extent\_info.extent\_size is size in bytes of this compressed extent.

# Data extents 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\_entity\_type values for the CompressionConfigurationBox are defined in Table 7. Generic compression is potentially applicable to other types of data besides samples and items (e.g., sample auxiliary information)

**Table 7 – Compressed entity types**

|  |  |
| --- | --- |
| Value | Description |
| 0 | the entity is the full item or sample |
| Image-related types | |
| 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 |
| Other types | |
| 5 | the entity is a single KLV-encoded key |
| other values | ISO/IEC reserved for future definition |

Entity ranges are specified in the same order as the entities themselves. For example, the entity ranges for image tiles are in the same order as the tiles as specified in Clause 5. Entity ranges for KLV-encoded keys are in the same order as the keys are found in the sample.

## 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 extent shall be the first byte of a CENC subsample.