ISO/IEC JTC 1/SC 29/WG 03 N0546

**ISO/IEC JTC 1/SC 29/WG 03  
MPEG Systems   
Convenorship: KATS (Korea, Republic of)**

**Document type:** Output Document

**Title:** Technology under Consideration on ISO/IEC 23008-12

**Status:** Approved

**Date of document:** 2022-05-05

**Source:** ISO/IEC JTC 1/SC 29/WG 03

**No. of pages:** 20 (with cover page)

**Email of Convenor:** young.L @ samsung . com

**Committee URL:** <https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg3>

**INTERNATIONAL ORGANIZATION FOR STANDARDIZATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 03 MPEG SYSTEMS**

**ISO/IEC JTC 1/SC 29/WG 03 N0546**

**May 2022, Virtual**

|  |  |
| --- | --- |
| **Title** | **Technology under Consideration on ISO/IEC 23008-12** |
| **Source** | **WG 03, MPEG Systems** |
| **Status** | **Approved** |
| **Serial Number** | **21439** |

# Abstract

This document collects following candidate technologies for the High Efficiency Image File Format (HEIF) (ISO/IEC 23008-12).

Table of Contents

[Abstract 1](#_Toc102637418)

[1. Carriage of Text Items 1](#_Toc102637419)

**[1](#_Toc102637420)****[Introduction](#_Toc102637420)** [1](#_Toc102637420)

**[2](#_Toc102637421)****[Proposal](#_Toc102637421)** [2](#_Toc102637421)

[2. Region annotations for image sequence or video tracks 7](#_Toc102637422)

**[1](#_Toc102637444)****[Proposal (updated from m59508)](#_Toc102637444)** [7](#_Toc102637444)

**[2](#_Toc102637445)****[Discussion](#_Toc102637445)** [10](#_Toc102637445)

[3. Matrix-based transformation for image items 10](#_Toc102637456)

[4. Signaling for pre-derived coded image items 10](#_Toc102637457)

*[Ed. (FD). It has been decided at MPEG#138 to move Camera and extrinsic camera parameters to HEIF CDAM.]*



# Carriage of Text Items

*[ Ed. (FD):* Open question from MPEG#137 meeting: *“we need to find a reasonably manageable/compact language: text/plain says too little (well, nothing) about styling, and full-on HTML seems a bit much”].*

This topic was discussed as [HEIF Issue#66](http://mpegx.int-evry.fr/software/MPEG/Systems/FileFormat/HEIF/-/issues/66) on the MPEG GitLab.

1. **Introduction**

Creating images with textual content such as captioning of images, memes and so on is very common and is done globally using various editing tools. Figure 1, shows an example meme image with plain text.



**Figure 1: Example of an meme image with plain text.**

The ISO/IEC 14496-30 standard on Timed text and other visual overlays in ISO base media file format specifies the carriage of timed text and subtitle streams in ISO BMFF tracks. However, neither ISO/IEC 14496-30 standard nor the HEIF standard specify the carriage of text items associated with an image item.

In MPEG 136, Nokia proposed a new item type called the text item for the carriage of textual content associated with an image item in m58143. The ISOBMFF group noted the proposal and provided the following suggestion:

We need to find a middle ground between plain text with no styling and the full complexity of something like HTML or SVG. We'd like to use but not overload the current technologies (such as overlay, MIME typed items, and so on).

Based on the above suggestions, this contribution proposes a mime type item for renderable text. The data in the mime type item is a renderable text.

1. **Proposal**

*Add following definitions in clause 3*

**3.1.X**

**text item**

*item* (3.1.27) whose data is the textual data.

**3.1.Y**

**renderable text item**

*a text item* (3.1.X) that includes possibly size, position, direction, language, font and styling and whose processing produces an output which can be visually rendered.

**3.1.Z**

**font item**

*item* (3.1.27) whose data is the fonts

*Update the subclause 6.5.2.1 as follows (changes highlighted in yellow)*

**6.5.2.1 Image spatial extents**

**6.5.2.2 Definition**

|  |  |
| --- | --- |
| Box type: | 'ispe' |
| Property type: | Descriptive item property |
| Container: | ItemPropertyContainerBox |
| Mandatory (per item): | Yes |
| Quantity (per item): | One |
|  |  |

The ImageSpatialExtentsProperty documents the width and height of the associated image item. Every image item shall be associated with one property of this type, prior to the association of all transformative properties.

The ImageSpatialExtentsProperty may be associated with items whose output can be visually rendered (e.g., renderable text items). When ImageSpatialExtentsProperty is associated with items whose output can be visually rendered, they document the visually rendered width and height of the data which is output from the associated item.

**6.5.2.3 Syntax**

aligned(8) class ImageSpatialExtentsProperty  
extends ItemFullProperty('ispe', version = 0, flags = 0) {  
 unsigned int(32) image\_width;  
 unsigned int(32) image\_height;  
}

**6.5.2.4 Semantics**

image\_width specifies the width of the reconstructed image in pixels, as specified in **Error! Reference source not found.**.

image\_height specifies the height of the reconstructed image in pixels, as specified in **Error! Reference source not found.**.

NOTE Item properties, such as decoder configuration or layer selection, can affect the reconstructed image. As a consequence, the width and height of the reconstructed image depend on the presence and content of such properties.

When ImageSpatialExtentsProperty is associated with items whose output can be visually rendered, the image\_width and image\_height specifies the visually rendered width and height, respectively of the data which is output from the associated item.

*Add the following subclause in Clause 6*

**6.A Text item and Renderable text item**

**6.A.1 Definition**

A text item is an item with item\_type value set to 'mime' and the data in the text item is text, for example, ‘html’ or ‘plain text’. The content\_type in ItemInfoEntry of the ItemInfoBox is set equal to the mime type of the data in the text item. Example values for content\_type field may include ‘text/html’ for html formatted text or ‘text/plain’ for plain text.

The text item is associated with the image item on which the textual data is displayed/rendered using an item reference of type 'cdsc' from the text item to the image item. A text item shall be associated with multiple image items only when all the associated image items have the same size. [NOTE: As an alternate to the use of the text item may be used as an overlay to the image item. However, this aspect needs to be further discussed]

The text item may be associated with the font item using an item reference of type 'font' from the text item to the font item. The font item carries the fonts used for rendering the text item.

The text item is associated with the ImageSpatialExtentsProperty which documents the visually rendered width and height of the data which is output from the text item.

The text item is associated with the TextLayoutProperty which documents the visually rendered size, position and language of the data which is output from the text item.

When a text item is not associated with any item or item property which documents possibly size, position, direction, language, font and styling for visual rendering, then, the data in the text item should contain the textual data together with possibly size, position, direction, language, font and styling for visual rendering of the text item and is called the renderable text item.

The renderable text data may be further encoded with either gzip or deflate or any other alogithm defined for content-encoding of Http/1.1. The encoding of renderable text data shall be defined by the content\_encoding parameter in ItemInfoEntry of the ItemInfoBox for the mime type text item.

If the renderable text data is encoded with any of the alogithm defined for content-encoding of Http/1.1, the data needs to be decoded before interpreting it as the mime type text item identified by the content\_type in ItemInfoEntry of the ItemInfoBox.

If the content\_encoding parameter in ItemInfoEntry of the ItemInfoBox has an empty string, then no content encoding is applied on the renderable text data.

The mime type item of renderable text only carries the text data required for rendering, however it does not provide any information on the display/layout conditions, for example the position, size and direction of the renderable text. Hence we propose a item property for the mime type text item which carries the information on the display/layout conditions.

[NOTE: As an alternate to the use of text layout property, the following approach may be used The ImageSpatialExtentsProperty to document the width and height,

The ImageOverlay to document the reference width and reference height and the position of the renderable text item

The ExtendedLanguageBox to document the language of the textual data. However, these aspect needs to be further discussed]

**6.B Text Layout Information**

**6.B.1 Definition**

Box type: 'txlo'

Property type: Descriptive item property

Container: ItemPropertyContainerBox

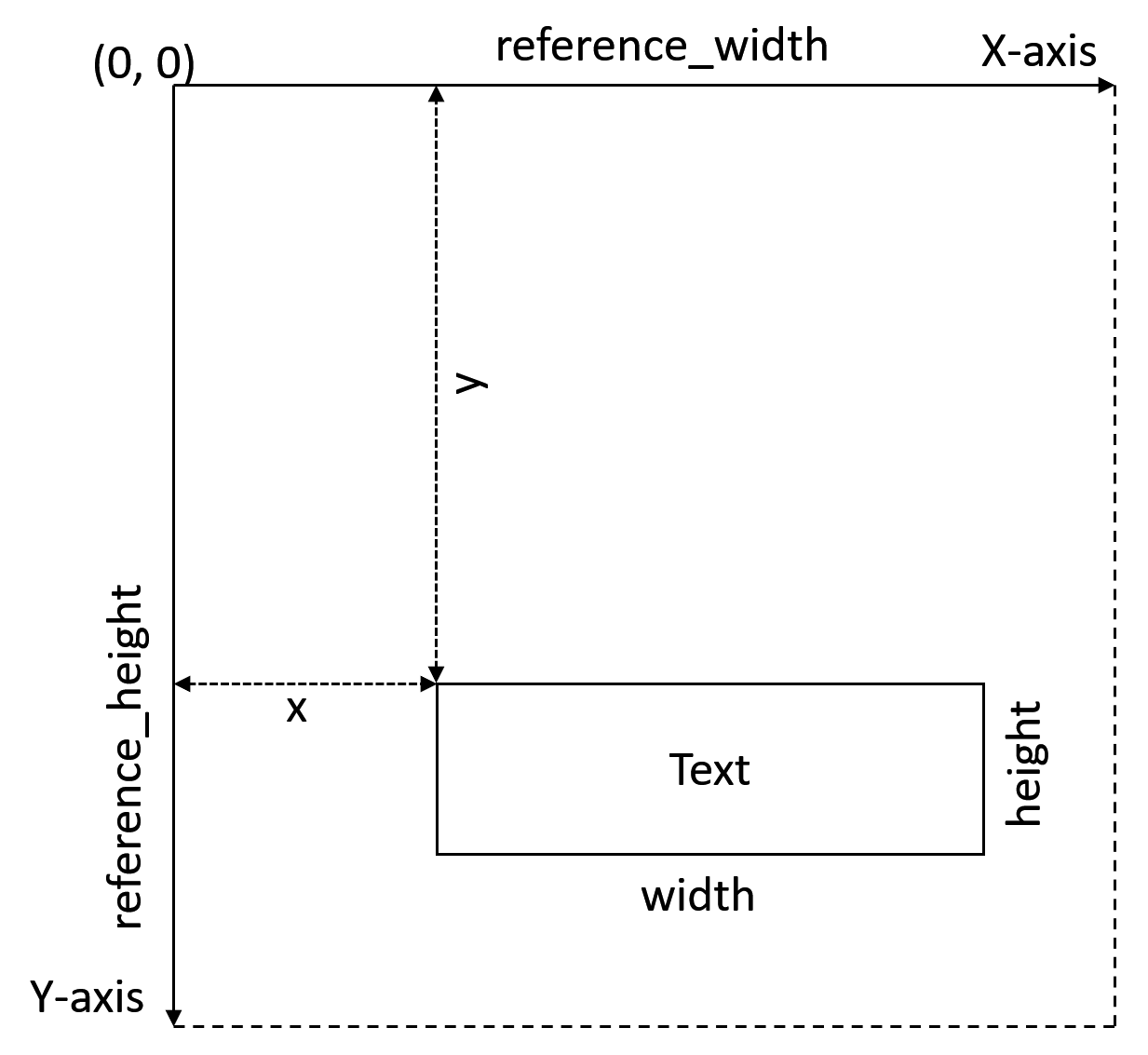
Mandatory (per item): No

Quantity (per item): One

The TextLayoutProperty documents the layout information of the associated text item. A text item shall be associated with one TextLayoutProperty prior to the association of all transformative properties.

The data in the TextLayoutProperty define the size, position and the language of the textual data to be displayed/rendered on the associated image item. The size and position information are used to display/render the textual data inside a reference space that is mapped to the image item with which the text item is associated after any transformative item property is applied to the image item.

The reference space is defined as a 2D coordinate system with the origin (0,0) located at the top-left corner and a maximum size defined by reference\_width and reference\_height; the x-axis is oriented from left to right and the y-axis from top to bottom. Figure 2, provides an illustration of text item in the reference space. The placement of textual data inside the associated image item is obtained after applying the implicit resampling caused by the difference between the size of the reference space and the size of the associated image item. If the text item has transformative item properties, then the implicit resampling shall be performed on the text item before the first of its transformative item properties is applied.



**Figure 2: An illustration of text item in reference space.**

**6.B.2 Syntax**

aligned(8) class TextLayoutProperty  
extends ItemFullProperty('txlo', version = 0, flags = 0) {  
 unsigned int (8) version = 0;   
 unsigned int (8) flags;   
 field\_size = ((flags & 1) + 1) \* 16;   
 unsigned int(field\_size) reference\_width;   
 unsigned int(field\_size) reference\_height;  
 signed int(field\_size) x;  
 signed int(field\_size) y;   
 utf8string language;   
}

**6.B.3 Semantics**

version shall be equal to 0.

(flags & 1) equal to 0 specifies that the length of the fields x, y, width, height is 16 bits. (flags & 1) equal to 1 specifies that the length of the fields x, y, width, height is 32 bits. The values of flags greater than 1 are reserved.

reference\_width, reference\_height specify, in pixel units, the width and height, respectively, of the reference space on which the text items are placed.

x, y specify the top, left corner of the text item relatively to the reference space. The value (x = 0, y = 0) represents the position of the top-left pixel in the reference space.

NOTE Negative values for the x or y fields enable to specify top-left corners that are outside the image. This can be useful for updating text items during the edition of an HEIF file.

language is a character string containing an RFC 5646 compliant language tag string, such as "en-US", "fr-FR", or "zh-CN“, representing the language of the text. When language is empty, the language is unknown/undefined.

**6.C Font item**

**6.C.1 Definition**

A font item is an item with the item\_type value set to 'mime' and the data in the font item are fonts for example ‘woff’ (Web Open Font Format) or ‘ttf’ (true type font). The content\_type in ItemInfoEntry of the ItemInfoBox is set equal to the mime type of the data in the font item. Example values for content\_type field may include ‘font/ttf’ for true type fonts or ‘font/woff’ for web open font format fonts.

The font item may be associated with the text item using an item reference of type 'font' from the text item to the font item.

The font data may be further encoded with either gzip or deflate or any other alogithm defined for content-encoding of Http/1.1. The encoding of font data shall be defined by the content\_encoding parameter in ItemInfoEntry of the ItemInfoBox for the font item.

If the font is encoded with any of the alogithm defined for content-encoding of Http/1.1, the data needs to be decoded before interpreting it as the mime type font item identified by the content\_type in ItemInfoEntry of the ItemInfoBox.

If the content\_encoding parameter in ItemInfoEntry of the ItemInfoBox has an empty string, then no content encoding is applied on the font data.

# Region annotations for image sequence or video tracks

*[Ed. (FD). It has been decided at MPEG#138 to move Region Annotations for image sequence and video tracks as proposed in m59508 to HEIF CDAM,* ***except the interpolate flag that may need further thoughts*** *(the yellow highlighted syntax and semantics ([comment on MPEG Gitlab](http://mpegx.int-evry.fr/software/MPEG/Systems/FileFormat/HEIF/-/issues/69" \l "note_60556)).]*

1. **Proposal (updated from m59508)**

*[Ed. (FD)This TuC only contains parts requiring further discussion (the interpolate flag in sample format for region tracks)]The HEIF CDAM defines the region track*

**X.X Region track and region annotations for an image sequence or video track**

**X.X.1 General**

A metadata track with a sample entry 'rgan' is a region track whose samples define one or more regions inside images carried in samples of an associated image sequence or video track (also denoted source track in the following).

*(… see HEIF CDAM for full text… )*

**X.X.3 Sample format**

**X.X.3.1 Definition**

This subclause defines the sample format for region track. A sample of a region track defines one or more regions.

**X.X.3.2 Syntax**

aligned (8) class RegionSample {  
 unsigned int field\_size = ((RegionTrackConfigBox.field\_length\_size & 1) + 1) \* 16;   
// this is a temporary, non-parsable variable  
 unsigned int(7)reserved;  
 unsigned int(1)interpolate;  
 unsigned int(16) region\_count;  
 for (r=0; r < region\_count; r++) {  
 unsigned int(32) region\_identifier;  
 unsigned int(8) geometry\_type;  
 if (geometry\_type == 0) {  
 // point  
 signed int(field\_size) x;  
 signed int(field\_size) y;  
 }  
 else if (geometry\_type == 1) {  
 // rectangle  
 signed int(field\_size) x;  
 signed int(field\_size) y;  
 unsigned int(field\_size) width;  
 unsigned int(field\_size) height;  
 }  
 else if (geometry\_type == 2) {  
 // ellipse  
 signed int(field\_size) x;  
 signed int(field\_size) y;  
 unsigned int(field\_size) radius\_x;  
 unsigned int(field\_size) radius\_y;  
 }  
 else if (geometry\_type == 3 || geometry\_type == 6) {  
 // polygon or polyline  
 unsigned int(field size) point\_count;  
 for (i=0; i < point\_count; i++) {  
 signed int(field\_size) px;  
 signed int(field\_size) py;  
 }  
 }  
 else if (geometry\_type == 4) {   
 // referenced mask  
 signed int(field\_size) x;  
 signed int(field\_size) y;  
 unsigned int(field\_size) width;  
 unsigned int(field\_size) height;  
 unsigned int(field\_size) track\_mask\_idx;  
 }  
 else if (geometry\_type == 5) {   
 // inline mask   
 signed int(field\_size) x;  
 signed int(field\_size) y;  
 unsigned int(field\_size) width;  
 unsigned int(field\_size) height;  
 unsigned int(8) mask\_coding\_method;   
 if (mask\_coding\_method != 0)   
 unsigned int(32) mask\_coding\_parameters;  
 bit(8) data[];  
 }  
 }  
}

**X.X.3.3 Semantics**

interpolate indicates the continuity in time of the successive samples. When true, the application may linearly interpolate values of the region geometries between the previous sample and the current sample. When false, there shall not be any interpolation of values between the previous and the current samples.

NOTE 1 When using interpolation, it is expected that the interpolated samples match the presentation time of the samples in the referenced source track. For instance, for each video sample of a video track, one interpolated region sample is calculated.

region\_count specifies the number of regions defined in the sample.

region\_identifier specifies the identifier of the region.

geometry\_type specifies the type of the geometry of a region. The following values for geometry\_type are defined:

0: the region is described as a point.

1: the region is described as a rectangle.

2: the region is described as an ellipse.

3: the region is described as a polygon.

4: the region is described as a mask defined in a referenced image item or in a sample of a referenced track.

5: the region is described as a mask defined inside the data of this sample.

6: the region is described as a polyline.

Other values are reserved.

x, y specify the coordinates of the point composing the region relatively to the reference space when its geometry is a point. x, y specify the top, left corner of the region relatively to the reference space when its geometry is a rectangle or a mask. x, y specify the centre of the region relatively to the reference space when its geometry is an ellipse. The value (x = 0, y = 0) represents the position of the top-left pixel in the reference space.

NOTE 2 Negative values for the x or y fields enable to specify points, top-left corners, and/or centres that are outside the image. This can be useful for updating region annotations during edition.

width, height specify, relatively to the reference space, the width and the height of the region when its geometry is a rectangle or a mask. When geometry\_type equals 4, the value 0 indicates that the corresponding width or height value is provided by the ImageSpatialExtentsProperty associated with the item containing the mask or the width or height of the track containing the mask. When geometry\_type does not equal 4, the value 0 is reserved.

radius\_x specifies, relatively to the reference space, the radius on the x-axis of the region when its geometry is an ellipse.

radius\_y specifies, relatively to the reference space, the radius on the y-axis of the region when its geometry is an ellipse.

point\_count is the number of points contained in a polygon or a polyline.

NOTE 3 A polygon specifying the geometry of a region is always closed and therefore there is no need to repeat the first point of the polygon as the ending point of the polygon.

px, py specify the coordinates of the points composing the polygon or the polyline relatively to the reference space. The value (px = 0, py = 0) represents the position of the top-left pixel in the reference space.

track\_mask\_idx specifies the index of the track reference of type 'mask' referring to the track from which to retrieve the mask to apply. The sample in that track from which mask data is retrieved is the one that is temporally aligned with the current sample in the source track or the nearest preceding one in the media presentation timeline. The first track reference has the index value 1; the value 0 is reserved.

mask\_coding\_method indicates the coding method applied on the mask contained in data. The following values are defined:

0: No mask encoding scheme is applied.

1: Mask is compressed with deflate() as defined in IETF RFC 1951.

Other values are reserved.

mask\_coding\_parameters indicates additional encoding parameters needed for successfully processing the coded mask data. When mask\_coding\_method is equal to 1, mask\_coding\_parameters indicates the number of bytes in the coded mask array data. The value of mask\_coding\_parameters is reserved when the value of mask\_coding\_method is greater than 1.

data contains the coded or uncompressed representation of a mask that contains the pixels for an inline mask in raster-scan order. Each pixel is represented using a single bit and 8 pixels are packed in one byte. Byte packing shall be in big-endian order. No padding shall be put at the end of each line if the width of the mask is not a multiple of 8 pixels. Only the last data byte shall be padded with bits set to 0.

1. **Discussion**

About the interpolate flag: The purpose is to avoid declaring a sample in the region track for each sample of the media track when regions are moving linearly between two positions. Imagine a sample A in the region track with a region at a starting position A and this region is moving linearly to the arrival position B nine samples later. Instead of declaring ten samples in the region track, you can only declare two samples, sample A with a duration corresponding to nine samples in the media track, followed by sample B providing the arrival position B. We should clarify that since the interpolate flag applies to all regions in the sample, the number of regions shall be the same in sample A and B.

# Matrix-based transformation for image items

*[[ Ed. (FD): MPEG#129: it was questioned:”* Should we also add ‘matrix’ as an image derivation in the HEIF? “. It was warned that “We would need to be clear about the meaning of outputs that don’t have horizontal and vertical sides; if that’s overlaid, the meaning is clear, but what if it’s supposed to be displayed?”*]]*

# Signaling for pre-derived coded image items

*Replace the clause 6.4.7 with the following text:*

**6.4.7** **Pre-derived coded images**

[Ed. (FD): In the following, differences with HEIF 2nd edition (w18310) are highlighted in blue]

If a coded image has been derived from others — for example, a composite HDR image derived from exposure-bracketed individual images, or a panorama derived from a set of images — then it shall be linked to those images by item references of type 'base'. Item references may be from the coded image to all images it derives from, or when unique IDs are used, from the coded image to all entity groups or images it derives from. When unique IDs are used, a to\_item\_ID value in the SingleItemTypeReferenceBox or SingleItemTypeReferenceBoxLarge is resolved to an item identifier whenever the embedding MetaBox contains an item with such identifier, and is resolved to an entity group identifier otherwise.

An image item including a 'base' item reference is referred to as a pre-derived coded image.

NOTE In this version of this document, the exact derivation process used to produce the image is not described.

[[Ed. (FD): At MPEG#129, it was commented that “The slight snag here is defining what it means when the entity group does NOT imply a single output (e.g. a slide show); what does pre-derivation mean? ]]

*Add the following clause as section 6.4.7.1:*

**6.4.7.1 Signaling of the derivation method for pre-derived coded image items**

A pre-derived coded image shall be linked to images it derives from by an item reference of type 'base' to the entity group containing all images the pre-derived coded images derives from. The grouping\_type of the EntityToGroupBox specifies the purpose of grouping and implicitly signals the type of the derivation operation which was applied to generate the pre-derived coded image.

[[Ed. (FM): At MPEG#126, it was commented that “we somehow need to indicate the derivation operation, rather than the nature of the input set”]]

[[Ed. (FD): At MPEG#129, it was commented that “We could allow a pre-derivation of the implied derivation of that entity group.”]]