INTERNATIONAL STANDARD

ISO/IEC 14496-16

Fourth edition
2011-11-01

AMENDMENT 3
2016-12-15

Information technology — Coding of audio-visual objects —

Part 16: Animation Framework eXtension (AFX)

AMENDMENT 3: Printing material and 3D graphics coding for browsers

Technologies de l'information — Codage des objets audiovisuels —

Partie 16: Extension du cadre d'animation (AFX)

AMENDEMENT 3: Représentation efficiente de maillages 3D à multiples attributs

Reference number
ISO/IEC 14496-16:2011/Amd.3:2016(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

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. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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).

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).

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

For an explanation on 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.

Amendment 3 to ISO/IEC 14496-16:2011 was prepared by ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Information technology — Coding of audio-visual objects —

Part 16:
Animation Framework eXtension (AFX)

AMENDMENT 3: Printing material and 3D graphics coding for browsers

Page 48, 4.3.6.4.2

Replace the following:

```
IndexedRegionSet {
  coord Coordinate {point [
    0 0 0, 1 0 0, 1 1 0, 0 1 0, 0 1 1, 0 0 1, 1 0 1, 1 1 1, 0.5 0.5 0
  ]}
  texCoord TextureCoordinate {point [
    0 0, 0.333 0, 0.667 0, 1 0, 0 1, 0.333 1, 0.667 1, 1 1, 0.667 0.5
  ]}
  region [
    Region {
      coordIndex [
        2 1 8 -1, 1 0 8 -1, 0 3 8 -1, 3 2 8 -1, 1 2 7 -1, 7 6 1 -1, 5 6 7 -1, 7 4 5 -1
      ]
      texCoordIndex [5 1 0 4 7 6 2 3 8]
      texture ImageTexture {url "../pix/136.png"}
    }
    Region {
      coordIndex [4 3 0 -1, 0 5 4 -1, 6 5 0 -1, 0 1 6 -1, 3 4 7 -1, 7 2 3 -1]
      texCoordIndex [4 7 6 5 1 0 3 2]
      texture ImageTexture {url "../pix/245.png"}
    }
  ]
}
With the following:
IndexedRegionSet {
[…]
    Region {
      coordIndex [
        2 1 8 -1, 1 0 8 -1, 0 3 8 -1, 3 2 8 -1, 6 5 0 -1, 0 1 6 -1, 5 6 7 -1, 7 4 5 -1
      ]
[…]
    }
    Region {
      coordIndex [4 3 0 -1, 0 5 4 -1, 1 2 7 -1, 7 6 1 -1, 3 4 7 -1, 7 2 3 -1]
[…]
    }
}
```

Page 48, 4.3.7

Replace the current 4.3.7 (which shall be renumberred to 4.3.8) with the following.

4.3.7 3D meshes with Printing Material Texture

4.3.7.1 General

The **IndexedPrintingRegionSet (IPRS)** node is based on the **IndexedRegionSet** described in 4.3.6 and it describes region-based printing material information which can be used in the 3D printing application. The main design concept of **IPRS** is that the representation shall be easy to the designer. Based on this concept, **IPRS** has adopted texture mapping method for describing the printing material information because it is very popular to the designer. Thanks to the region-based texture mapping, per face or region or global material mapping is possible.

4.3.7.2 IndexedPrintingRegionSet node

4.3.7.2.1 Node interface

IndexedPrintingRegionlSet {

eventIn	**MFInt32**	**set_colorIndex**		
eventIn	**MFInt32**	**set_coordIndex**		
eventIn	**MFInt32**	**set_normalIndex**		
eventIn	**MFInt32**	**set_texCoordIndex**		
exposedField	**SFString**	**unit**	**#mm, cm, m, inch**	
exposedField	**SFFloat**	**minimumVerticalResolution**	**#unit is unit**	
exposedField	**SFNode**	**color**	**NULL**	
exposedField	**SFNode**	**coord**	**NULL**	
exposedField	**SFNode**	**normal**	**NULL**	
exposedField	**SFNode**	**texCoord**	**TRUE**	
field	**SFBool**	**ccw**	**[]**	**#[-1,inf)**
field	**MFInt32**	**colorIndex**	**TRUE**	
field	**SFBool**	**colorPerVertex**	**TRUE**	
field	**SFBool**	**convex**	**[]**	
field	**MFNode**	**printingRegions**	**0**	**#[0,inf)**
field	**SFFloat**	**creaseAngle**	**[]**	**#[-1,inf)**
field	**MFInt32**	**normalIndex**		
field	**SFBool**	**normalPerVertex**	**TRUE**	
field	**SFBool**	**solid**	**TRUE**	
field	**MFInt32**	**texCoordIndex**	**[]**	**#[0,inf)**

}

4.3.7.2.2 Functionality and semantics

An **IPRS** node has exactly the same fields as an **IRS** one, except for the physical size information of printout, and ***region*** field, which has been replaced by ***printingMaterialRegion***.

4.3.7.3 PrintingRegion node

4.3.7.3.1 Node interface

PrintingRegion {

exposedField	**MFInt32**	**printingMaterialType**	**NULL**	
exposedField	**SFFloat**	**surfaceThickness**	**#unit is unit**	
exposedField	**SFNode**	**color**	**NULL**	
exposedField	**SFNode**	**normal**	**NULL**	
exposedField	**SFNode**	**texCoord**	**NULL**	
exposedField	**SFNode**	**colorTexture**	**NULL**	
exposedField	**SFNode**	**printingMaterialTexture**	**NULL**	
exposedField	**SFNode**	**textureTransform**	**NULL**	
field	**MFInt32**	**colorIndex**	**[]**	**#[-1,inf)**
field	**MFInt32**	**coordIndex**	**[]**	**#[-1,inf)**
field	**MFInt32**	**normalIndex**	**[]**	**#[-1,inf)**
field	**MFInt32**	**texCoordIndex**	**[]**	**#[0,inf)**
field	**MFInt32**	**printingMaterialTexCoordIndex**	**[]**	**#[0,inf)**

}

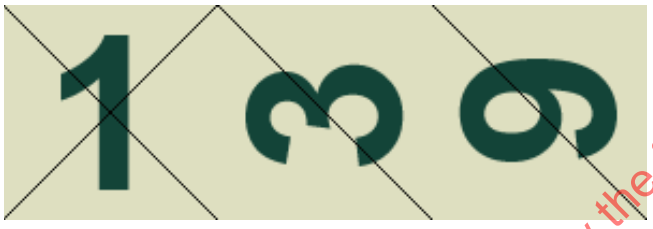
4.3.7.3.2 Functionality and semantics

field name	semantic
unit	It has single string value which shall be one of the length unit defined in the ISO/IEC 23005-6 (UnitTypeCS). It defines the unit of the ***coord*** field. When this value is set as mm, all the ***coord*** values are interpreted as mm (millimetre).
surfaceThickness	It defines the suggested surface thickness of the model. For example, a cylinder model with 5 ***surfaceThickness*** is interpreted as "Print the cylinder with 5 mm surface and leave the inside empty." Here, the unit for ***surfaceThickness*** is the same as ***unit*** files.
minimumVerticalResolution	It defined the resolution for the acquisition process. When the model is designed based on the scanner, the scanner resolution is mentioned here. When this field is set as 0.1, the 3D printing resolution which is smaller than ***minimumVerticalResolution*** does not make sense. In this way, the printing resolution may be estimated. Here, the unit for ***minimumVerticalResolution*** is the same as ***unit*** files.
printingRegions	It has **PrintingRegion** node which defined the physical material characteristics of print-out. It has multiple nodes. When a single element is provided, the whole model is considered as one region.
printingMaterialType	It has multiple integers that defines the printing materials of print-out as a reference to a classification defined in ISO/IEC 23005-6 (PrintingMaterialCS). When a single material is provided, the whole region is printed as one material.

field name	semantic
colorTexture	It defines the color texture used in each region for rendering purpose.
printingMaterialTexture	It defines the printing material texture map used in each region. It shall be lossless gray image format such as PNG because the lossy compression results in a misunderstanding on the printing material information. And the values in the image are restricted by ***printingMaterialType***. All values in the printing material texture are one value in the ***printingMaterialType*** array. For example, when the ***printingMaterialType*** is provided as [0, 1, 2], the ***printingMaterialTexture*** shall has values among 0, 1, and 2. When the array length of ***printingMaterialType*** is 1, the ***printingMaterialTexture*** is null because this region is printed as a single material with ***printingMaterialType***.
texCoordIndex	It defines the texture coordinate indexes of ***colorTexture***.
printingMaterialTexCoordIndex	It defines the texture coordinate indexes of ***printingMaterialTexture***. When this is null, ***texCoordIndex*** is used as ***printingMaterialTexCoordIndex***.

4.3.7.4 Examples

The following examples are based on the IRS representation described in 4.3.6.4.3. It has two printingRegion and each printingRegion has *colorTexture* and *printingMaterialTexture* as illustrated in Figure 25.



a) colorTexture(136.png)

1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	0	0	0	0	2	1	0	0	0	0	0
0	0	1	1	1	1	0	0	0	0	2	2	1	1	0	0	0	0
0	0	0	1	1	1	0	0	0	2	2	2	1	1	1	0	0	0
0	0	0	0	1	1	0	0	2	2	2	2	1	1	1	1	0	0
0	0	0	0	0	1	0	2	2	2	2	2	1	1	1	1	1	0

b) printingMaterialTexture(136_material.png)

c) colorTexture(245.png)

2	2	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
1	2	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
1	1	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	2	2	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	2	0	0	0	0	0	0	0	0	0	0	0	0

d) printingmaterialTexture(245_material.png)

Figure 25 — IPRS example with two printingRegion;

In this example, three materials [0, 1, 2] are used thus the values in the printingMaterialTexture are restricted by 0, 1 and 2.

```
IndexedPrintingRegionSet{
  unit "mm"
  minimumVerticalResolution 0.1
  coord Coordinate {
    point [ 0 0 0, 1 0 0, 1 1 0, 0 1 0, 0 1 1, 0 0 1, 1 0 1, 1 1 1, 0.5 0.5 0 ]
  }
  texCoord TextureCoordinate {
    point [ 0 0, 0.333 0, 0.667 0, 1 0, 0 1, 0.333 1, 0.667 1, 1 1, 0.1667 0.5 ]
  }
  printingRegions [
    PrintingRegion {
       surfaceThickness 10.0
       printingMaterialType [0, 1, 2]
       coordIndex [ 2 1 8 -1, 1 0 8 -1, 0 3 8 -1, 3 2 8 -1, 6 5 0 -1, 0 1 6 -1, 5 6 7 -1,
7 4 5 -1 ]
       texCoordIndex [5 1 0 4 7 6 2 3 8]
       colorTexture            ImageTexture  {  url "../pix/136.png"  }
       printingMaterialTexture ImageTexture  {  url "../material/136_material.png" }
    }
    PrintingRegion {
      surfaceThickness 15.0
      printingMaterialType [0, 1, 2]
      coordIndex [4 3 0 -1, 0 5 4 -1, 1 2 7 -1, 7 6 1 -1, 3 4 7 -1, 7 2 3 -1]
      texCoordIndex [4 7 6 5 1 0 3 2]
      colorTexture            ImageTexture   { url "../pix/245.png" }
      printingMaterialTexture  ImageTexture   { url "../material/245_material.png" }
    }
  ]
}
```

Page 273, Clause 6

Insert the following clause.

6 Web3D Coding

6.1 General

This document describes a scene representation using a JSON schema. The reason for choosing JSON is the native support by modern browsers and easiness of integration with javascript. The JSON schema is used to connect the object 2graph elements, MPEG AFX tools, images and shaders. The main components of the library (see Figure 82) are the JSON parser and the GraphicsCodec, whereas the GraphicsCodec contains both the SC3DMC and BBA decoders. The input of the library is the JSON description file. The JSON Parser analyses its data and calls the appropriate decoders, the SC3DMC decoder for the mesh data and BBA decoder for the animated data. The output of the decoders is then used to initialize the corresponding glIndexedFaceSet structures that are then used by the WebGL engine to render the scene.

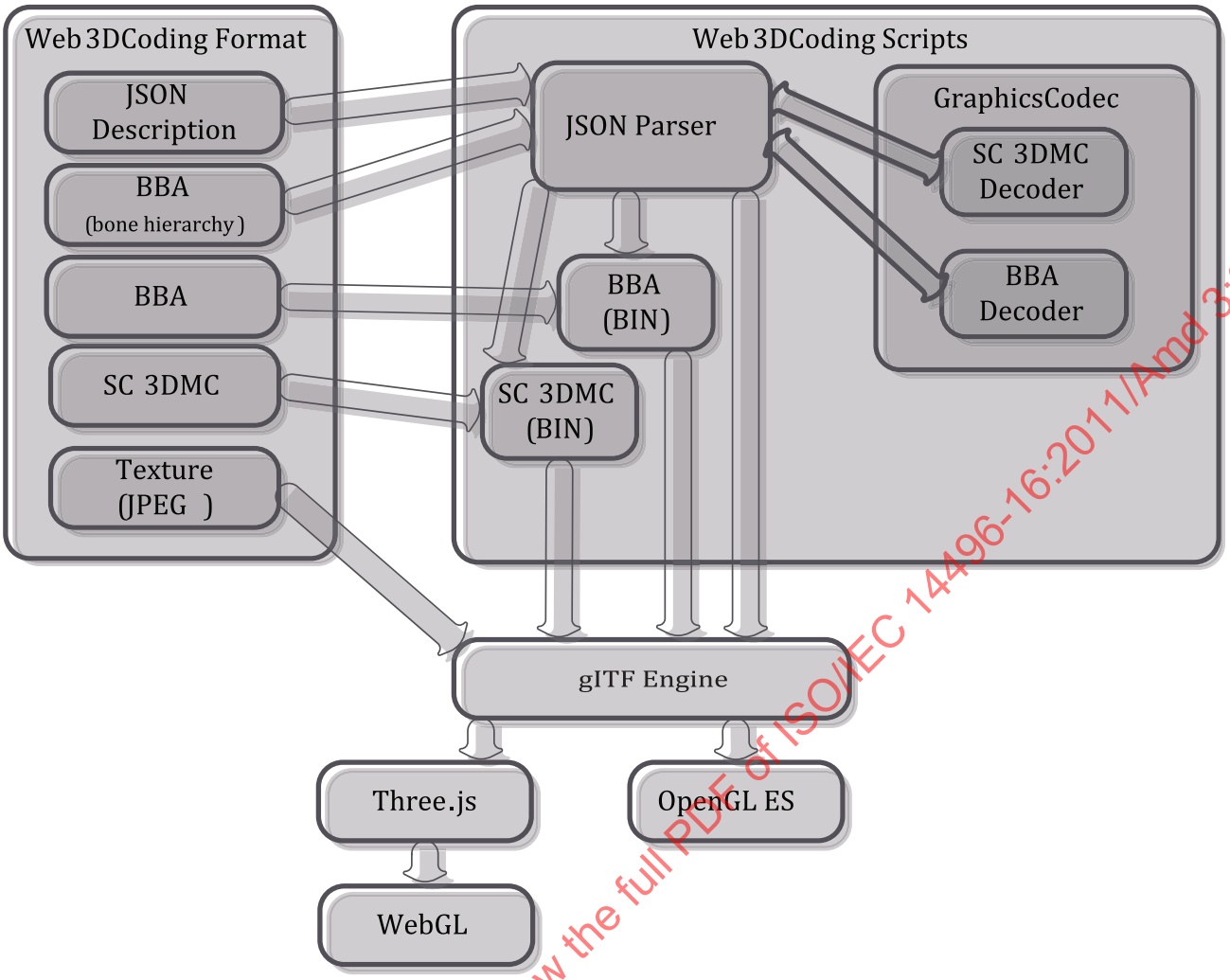


Figure 82 — Overview of the Web3DCoding architecture

6.2 Scope

Define a JSON schema that allows connections between object graph elements to MPEG 4 AFX compression tools (SC3DMC and BBA), image compression tools (JPEG, JPEG2000 and PNG) and shaders.

6.3 JSON Schema

```
{
  "type": "object",
  "properties": {
    "object": {
        "type": "object",
        "properties": {
            "name": {
                 "type": "string",
                 "description" : "The name of the object. The name has to be unique in
                 the object
                 definition"
            },

            "shapes": {
                "type": "array",
                "items": {
                     "type": "object",
                     "properties" : {
                         geometry_filename" : {
                            "type" : "string",
```

```
                "description" : "An URL to the file containing the 3D mesh
                compressed (MPEG-SC3DMC) data for the shape"
},

"transform": {
    "type" : "array",
    "items" : {
        "type" : "number"
    },
    "description" : "An Array of values representing the transformation
    matrix specifying the transformation of the shape relative to the object
    center."
},

"appearance" : {
    "type" : "object",
    "properties" : {
        "lineProperties" : {
            "type" : "object",
            "properties" : {
                "applied" : {
                    "type" : "boolean",
                    "description" : "Specifies if the additional properties
                    shall be applied to all the line geometry."
                },
                "linetype" : {
                   "type" : "number",
                   "description" : "Selects a line pattern as defined in the
                   Table 77. Line type 1 is the default. X3D. If a
                   linetype that is not supported is requested, linetype 1
                   shall be used."
                },
                "linewidthScaleFactor" : {
                    "type" : "number",
                    "description" : "is a multiplicative value that scales a
                    browser dependent nominal linewidth by the linewidth scale
                    factor. This resulting value shall then be mapped to the
                    nearest available line width. A value less than or equal
                    to zero refers to the minimum available line width.."
                }
        },
                "description" : "Specifies additional properties to be applied
                to all line geometry."
        },

        "materials" : {
            "type" : "object",
            "properties" : {
                "diffuseColor" : {
                    "type" : "array",
                    "items" : {
                        "type" : "number"
                    },
                    "description" : "An array of values, reflects all light
                    sources depending on the angle of the surface with
                    respect to the light source. The more directly the
                    surface faces the light, the more diffuse light reflects"
                },
                "emissiveColor" :{
                    "type" : "array",
                    "items" : {
                        "type" : "number"
                    },
                    "description" : "An array of values, represents the
                    Models <<glowing>> objects. This can be useful for
                    displaying pre-lit models (where the light energy of the
                    room is computed explicitly), or for displaying
                    scientific data."
                },
                "shininess" : {
                    "type" : "number",
```

```
                    "description" : "Determine the specular highlights (e.g.,
                    the shiny spots on an object). Lower shininess values
                    produce soft glows, while higher values result in
                    sharper, smaller highlights"
                },
                "specularColor" : {
                    "type" : "array",
                    "items" : {
                        "type" : "number"
                    },
                    "description" : "An array of values. When the angle from
                    the light to the surface is close to the angle from the
                    surface to the viewer, the specularColor is added to the
                    diffuse and ambient colour calculations."
                    },
                    "transparent" : {
                        "type" : "boolean",
                        "description" : "Defines whether this material is
                        transparent"
                    },
                    "opacity" : {
                        "type" : "number",
                        "description" : "Specifies how 'clear' an object is,
                        with 0.5 being a bit transparent"
                    }
                },
            "description" : "An array of materials, one for each sub-shape of
            the shape."
        },

        "shaders" :{
            "type" : "array",
            "items" : {
                "type" : "object",
                "properties" : {
                    "activate" : {
                        "type" : "boolean",
                        "description" : "A Specifies if the current shader is
                        active or not."
                    },
                    "isSelected" : {
                        "type" : "boolean",
                        "description" : "Indicates that this shader instance
                        is the one selected for use by the browser."
            },
            "isValid" : {
                "type" : "boolean",
                "description" : "Indicates whether the current shader objects
                can be run as a shader program."
            },
            "language" : {
                "type" : "string",
                "description" : "Indicates to the browser which shading
                language is used for the source file. This field may be used
                as a hint for the browser if the shading language is not
                immediately determinable from the source (e.g., a generic
                MIME type of text/plain is returned). A browser may use this
                field for determining which node instance will be selected
                and to ignore languages that it is not capable of supporting.
                Three basic language types are defined for this specification
                and others may be optionally supported by a browser.."
            },
            "shader_fileName" : {
                "type" : "string",
                "description" : "An URL pointing to the shader definition
                file in the specified language."
            }
        },
        "required" : ["activate", "isSelected", "isValid", "language",
        "shader_fileName"]
    }
```

```
        },

        "texture_fileName" : {
            "type" : "string",
            "description" : "An URL pointing to the texture for the material."
        },

        "textureTransform" : {
            "type" : "object",
            "properties" : {
                "center" : {
                    "type" : "array",
                    "items" : {
                        "type" : "number"
                    },
                    "description" : "Specifies a translation offset in texture
                    coordinate space about which the rotation and scale fields are
                    applied."
                },
                "rotation" : {
                    "type" : "number",
                    "description" : "Specifies a rotation in radians of the texture
                    coordinates
                    about the center point after the scale has been applied. A
                    positive rotation value makes the texture coordinates rotate
                    counterclockwise about the centre, thereby rotating the appearance
                    of the texture itself clockwise."
                },
                "scale" : {
                    "type" : "number",
                    "description" : "Specifies a scaling factor in S and T of the
                    texture coordinates about the center point."
                },
                "translation" : {
                    "type" : "array",
                    "items" : {
                        "type" : "number"
                    },
                    "description" : "Specifies a translation of the texture
                    coordinates."
                }
            }
        }

    },
    "description" : "specifies the visual properties of geometry"
},
"anm" : {
    "type" : "array",
    "items" : {
        "type" : "object",
        "properties" : {
            "numFrames" : {
                "type" : "number",
                "description" : "Specifies the number of frames for the current set
                of properties."
            },
            "localCoordinates" : {
                "type" : "boolean",
                         "description" : "Specifies if the bone transforms are
                         represented in local coordinates."
                     },
                     "skeleton" : {
                         "type" : "array",
                         "items":{
                             "type" : "object",
                             "properties" : {
                                 "numBones" : {
                                     "type" : "number",
                                     "description" : "Specifies the number of bones in
                                     the current skeleton."
```

```
                            },
                            "bones" : {
                                "type" : "array",
                                "items" : {
                                    "type" : "object",
                                    "properties" : {
                                        "boneID" : {
                                            "type" : "number",
                                            "description" : "Specifies the unique
                                            id of the bone."
                                        },
                                        "parentID" : {
                                            "type" : "number",
                                            "description" : "Specifies the parent
                                            bone unique id of the bone."
                                        },
                                        "boneTransform" : {
                                            "type" : "array",
                                            "items" : {
                                                "type" : "number"
                                            },
                                            "description" : "Array of values
                                            representing the transform matrix of
                                            size 4x4 represented as 16 values
                                            array."
                                        }
                                    }

                            },
                            "description" : "Array of objects of type <<bones>>."
                        }

                    }

                },
                "description" : "Array of objects of type <<skeleton>> that
                specifie the set of bones connections and hierarchical
                relations."
            }
        },
        "required" : ["numFrames", "localCoordinates", "skeleton"]
        },
        "description" : "An optional field describing the connection between the
        bones and the vertices."
    },

    "animation" :{
        "type" : "object",
        "properties" : {
            "bba_streams" : {
                "type" : "array",
                "items" : {
                    "type" : "object",
                    "properties" : {
                        "name" : {
                            "type": "string",
                            "description" : "A name for the animation. It has to
                            be unique for the shape object."
                        },
                        "animation_fileName" : {
                            "type" : "string",
                            "description" : "An URL pointing to the BBA file
                            that corresponds to the animation."
                        }
                    },
                    "required" : ["name", "animation_fileName"]
                },
                "description" : "An array of BBA animation streams associated
                with the shape."
            }
        },
```