Official specification of A3D format

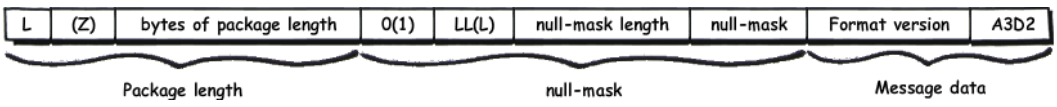
- Introducing
- File structure
    - Package length
    - Null-mask
        - Short null-mask
        - Long null-mask
    - Message data
        - Representation of arrays and strings.
        - Version
        - A3D2
        - A3D2AmbientLight
        - A3D2AnimationClip
        - A3D2Track
        - A3D2Box
        - A3D2CubeMap
        - A3D2Decal
        - A3D2DirectionalLight
        - A3D2Image
        - A3D2IndexBuffer
        - A3D2Joint
        - A3D2Map
        - A3D2Material
        - A3D2Mesh
        - A3D2Object
        - A3D2OmniLight
        - A3D2SpotLight
        - A3D2Sprite
        - A3D2Skin
        - A3D2JointBindTransform
        - A3D2Keyframe
        - A3DMatrix
        - A3D2Surface
        - A3D2Transform
        - A3D2VertexBuffer
        - A3D2VertexAttributes
        - ByteBuffer
            - Structure
            - Example

Introducing

A3D - a format developed by the company AlternativaPlatform, designed to hold three-dimensional scenes in binary files. The binary presentation can transmit data without any conversion, greatly accelerating the speed of transmission.
This specification describes the general characteristics that should be considered when working with files in A3D.

File structure

File in A3D format is a package. The maximum size of data that can be packed in the package - 2147483648 bytes (packed in a situation with a restriction imposed after packaging). The package has the following structure:



Package length

L-flag of the size of packet length (1 bit)

L	Description
0	packet length contains the remaining 6 bits of the flag byte and the subsequent byte (14 bitonly), the second bit is specified on the package sign.
1	A packet length contained in the remaining 7 bits of the TI flag byte and the next 3 bytes (31bits). Packing takes an attribute to "true"

Z-flag package, indicated at L = 0 (1 bit)

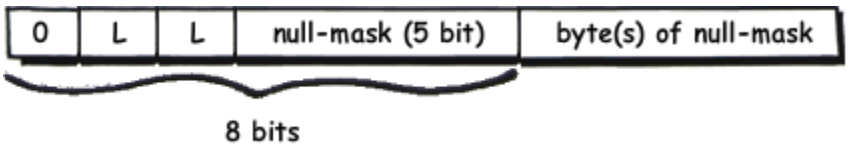
Z	Description
0	data packet is not packed
1	data packet is packed (gzip)

Null-mask

Bit mask, is each bit which determines the presence or absence of null objects in the message. Mask length in bytes.

Short null-mask

Encodes a zero mask to 29-bit. The first bit of the first byte is set to 0.

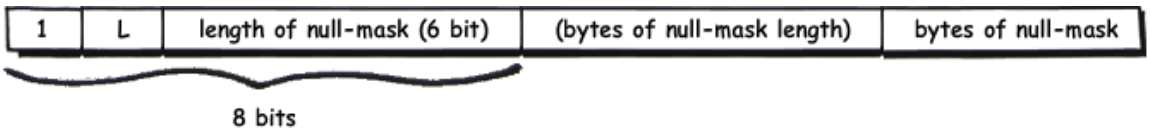


LL-length of null-mask (2 bits)

LL	Description
00	null-mask is found in the remaining 5 bits of the first byte (of 5 bits)
01	null-mask is found in the remaining 5 bits of the first byte and the subsequent byte (13 bits total)
10	null-mask is found in the remaining 5 bits of the first byte and the next 2 bytes (21 bits total)
11	null-mask is found in the remaining 5 bits of the first byte and the next 3 bytes (29 bits total)

Long null-mask

Encodes a zero mask to 33 554 432 bits. The first bit of the first byte is set to 1.



L-flag size masks length (1 bit)

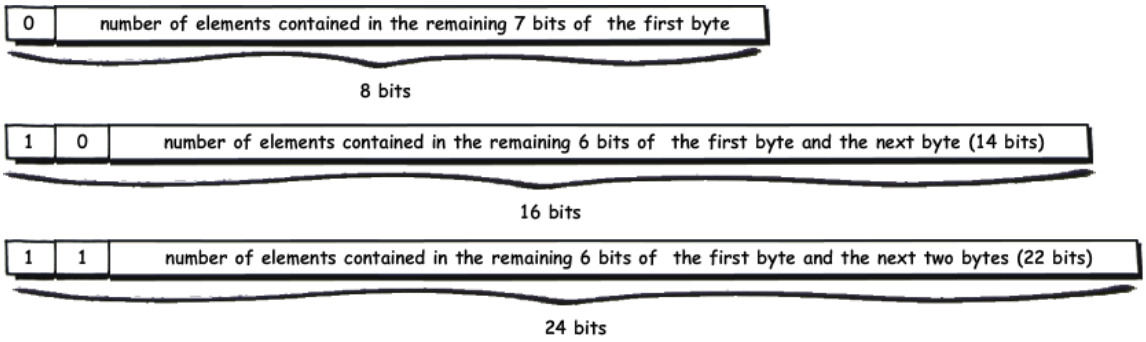
L	Description
0	length of the mask contained in the remaining 6 bits of the first byte (6 bits total)
1	length of the mask contained in the remaining 6 bits of the first byte and the next two bytes (22 bits total)

Message data

Representation of arrays and strings.

Strings in A3D - it's the arrays of characters in UTF-8. The overall structure of the coding arrays / lines:

The maximum number of elements is 4194304 of characters per line - 2097152.
There are 3 types of coding elements:



These items are stored in the properties.

Version

Field	Type	Size (bytes)	Additionally
Major version	ushort	2	2
Minor version	ushort	2	0

A3D2

One of the main requirements is to comply with the format logic location of properties. They are specified in the order shown in the following table:

Property	Type
ambientLights	array of A3D2AmbientLight

animationClips	array of A3D2AnimationClip
animationTracks	array of A3D2Track
boxes	array of A3D2Box
cubeMaps	array of A3D2CubeMap
decals	array of A3D2Decal
directionalLights	array of A3D2DirectionalLight
images	array of A3D2Image
indexBuffers	array of A3D2IndexBuffer
joints	array of A3D2Joint
maps	array of A3D2Map
materials	array of A3D2Material
meshes	array of A3D2Mesh
objects	array of A3D2Object
omniLights	array of A3D2OmniLight
spotLights	array of A3D2SpotLight
sprites	array of A3D2Sprite
skins	array of A3D2Skin
vertexBuffers	array of A3D2VertexBuffer

Properties marked as bold are required and must be specified. The other properties are optional.

If the optional property is not specified, then null-mask it should be noted 1 (1 bit).
If the optional property is specified, then null-mask it should be noted 0 (1 bit).

A3D2AmbientLight

See also A3D2

Property	Type	Size (bytes)	Additionally
boundBoxId	int	4	A3D2Box
color	uint	4	
id	int64	8	
intensity	float	4	
name	String		Array of char (UTF-8)
parentId	int64	8	
transform	A3D2Transform		
visible	bool	1	

A3D2AnimationClip

See also A3D2, A3D2Track

Property	Type	Size (bytes)	Additionally
id	int	4	
loop	bool	1	
name	String		Array of char (UTF-8)
objectIDs	array of int64		
tracks	array of int	4	

A3D2Track

See also A3D2, A3D2AnimationClip

Property	Type	Size (bytes)	Additionally
id	int	4	
keyframes	array of A3D2Keyframe		
objectName	String		Array of char (UTF-8)

A3D2Box

See also A3D2Object

Property	Type	Size (bytes)	Additionally
box	array of float		[minX, minY, minZ, maxX, maxY, maxZ]
id	int	4	

A3D2CubeMap

See also A3D2Material

Property	Type	Size (bytes)
backId	int	4
bottomId	int	4
frontId	int	4
id	int	4
leftId	int	4
rightId	int	4
topId	int	4

A3D2Decal

See also A3D2

Property	Type	Size (bytes)	Additionally
boundBoxId	int	4	A3D2Box
id	int64	8	
indexBufferId	int	4	
name	String		Array of char (UTF-8)
offset	float	4	
parentId	int64	8	
surfaces	array of A3D2Surface		
transform	A3D2Transform		
vertexBuffers	array of int		
visible	bool	1	

A3D2DirectionalLight

See also A3D2

Property	Type	Size (bytes)	Additionally
boundBoxId	int	4	A3D2Box
color	uint	4	
id	int64	8	
intensity	float	4	
name	String		Array of char (UTF-8)
parentId	int64	8	
transform	A3D2Transform		
visible	bool	1	

A3D2Image

See also A3D2Material, A3D2Map

Property	Type	Size (bytes)
id	int	4
url	String	Array of char (UTF-8)

A3D2IndexBuffer

See also A3D2

Property	Type	Size (bytes)	Additionally
byteBuffer	array of byte		each index uses 2 bytes (little-endian)

id	int	4	
indexCount	int	4	max 524287

A3D2Joint

See also A3D2

Property	Type	Size (bytes)	Additionally
boundBoxId	int	4	A3D2Box
id	int64	8	
name	String		Array of char (UTF-8)
parentId	int64	8	
transform	A3D2Transform		
visible	bool	1	

A3D2Map

See also A3D2Material

Property	Type	Size (bytes)
channel	ushort	2
id	int	4
imageId	int	4

A3D2Material

See also A3D2Surface

Property	Type	Size (bytes)
diffuseMapId	int	4
glossinessMapId	int	4
id	int	4
lightMapId	int	4
normalMapId	int	4
opacityMapId	int	4
reflectionCubeMapId	int	4
specularMapId	int	4

A3D2Mesh

See also A3D2

Property	Type	Size (bytes)	Additionally
boundBoxId	int	4	A3D2Box
id	int64	8	
indexBufferId	int	4	
name	String		Array of char (UTF-8)
parentId	int64	8	
surfaces	array of A3D2Surface		
transform	A3D2Transform		
vertexBuffers	array of int		
visible	bool	1	

A3D2Object

See also A3D2

Property	Type	Size (bytes)	Additionally
boundBoxId	int	4	A3D2Box
id	int64	8	
name	String		Array of char (UTF-8)
parentId	int64	8	
transform	A3D2Transform		
visible	bool	1	

A3D2OmniLight

See also A3D2

Property	Type	Size (bytes)	Additionally
attenuationBegin	float	4	
attenuationEnd	float	4	
boundBoxId	int	4	A3D2Box
color	uint	4	
id	int64	8	
intensity	float	4	
name	String		Array of char (UTF-8)
parentId	int64	8	
transform	A3D2Transform		

visible	bool	1	

A3D2SpotLight

See also A3D2

Property	Type	Size (bytes)	Additionally
attenuationBegin	float	4	
attenuationEnd	float	4	
boundBoxId	int	4	A3D2Box
color	uint	4	
falloff	float	4	
hotspot	float	4	
id	int64	8	
intensity	float	4	
name	String		Array of char (UTF-8)
parentId	int64	8	
transform	A3D2Transform		
visible	bool	1	

A3D2Sprite

See also A3D2

Property	Type	Size (bytes)	Additionally
alwaysOnTop	bool	1	
boundBoxId	int	4	A3D2Box
height	float	4	
id	int64	8	
materialId	int	4	
name	String		Array of char (UTF-8)
originX	float	4	
originY	float	4	
parentId	int64	8	
perspectiveScale	bool	1	
rotation	float	4	
transform	A3D2Transform		
visible	bool	1	
width	float	4	

A3D2Skin

See also A3D2

Property	Type	Size (bytes)	Additionally
boundBoxId	int	4	A3D2Box
id	int64	8	
indexBufferId	int	4	
jointBindTransforms	array of A3D2JointBindTransform		
joints	array of int64		
name	String		Array of char (UTF-8)
numJoints	array of ushort		
parentId	int64	8	
surfaces	array of A3D2Surface		
transform	A3D2Transform		
vertexBuffers	array of int	4	
visible	bool	1	

Each element of array numJoints is the number of bones on a surface. Bones indexes are stored in an array of joints. For example, if there are 2 surfaces in skin: the first has 4 bones with ids 0,2,3,4, and second has 3 bones with ids 1,2,5. Respectively, in the vector are the following joints: [0,2,3,4,1,2,5], and in the vector numJoints are the following: [4,3]

A3D2JointBindTransform

See also A3D2Skin

Property	Type	Size (bytes)	Additionally
bindPoseTransform	A3D2Transform		
id	int64	8	bone id

A3D2Keyframe

See also A3D2, A3D2Track

Property	Type	Size (bytes)	Additionally
time	float	4	
transform	A3D2Transform		

A3DMatrix

See also A3D2Transform

Property	Type	Size (bytes)

a	float	4
b	float	4
c	float	4
...		
l	float	4

A3D2Surface

See also A3D2Mesh, A3D2Material

Property	Type	Size (bytes)
indexBegin	int	4
materialId	int	4
numTriangles	int	4

A3D2Transform

See also A3D2Object

Property	Type
matrix	A3DMatrix

A3D2VertexBuffer

See also A3D2, A3D2VertexAttributes

Property	Type	Size (bytes)	Additionally
attributes	array of int		A3D2VertexAttributes
byteBuffer	ByteBuffer		little-endian float
id	int	4	
vertexCount	ushort	2	

A3D2VertexAttributes

Enumerated type
POSITION = 0, size at byteBuffer (3 float fields - x,y,z)
NORMAL = 1, size at byteBuffer (3 float fields - x,y,z)
TANGENT4 = 2, size at byteBuffer (4 float fields - tangent.x, tangent.y, tangent.z, binormalDirection)
JOINT = 3, size at byteBuffer (4 float fields - jointA.index, jointA.weight, jointB.index, jointB.weight)
TEXCOORD = 4, size at byteBuffer (2 float fields - u, v)

ByteBuffer

Structure

vertex:

Each attribute has a dimension which corresponds to the type attribute.
Order, type and number of attributes specified as an array attributes.

Example

There are 3 vertexes. For each of them are set position, normal and texture coordinates.
1st vertex:

- position = 10 20, 30
- normal = 0, 0, 1
- texture coordinates 0, 0

2nd vertex:

- position = 15, 20, 30
- normal = 0, 0, 1
- texture coordinates 0, 1

3rd vertex:

- position = 30, 0, 20
- normal = 0, 0, 1
- texture coordinates 1, 1

```
attributes = [0,1,4]

vertexCount = 3

byteBuffer = [[10,20,30] [0,0,1] [0,0]]  [[15,20,30] [0,0,1] [0, 1]]  [[30,0,20] [0,0,1] [1,1]]
```

As a result byteBuffer will looks like:
10,20,30,0,0,1,0,0,15,20,30,0,0,1,0,1,30,0,20,0,0,1,1,1
each item is recorded in format litle endian float