```
.PBA
#define PBA_VERSION 0x0002
#define BASE CHUNK PBA (0x300)
#define CHUNK_PBA_MAIN (0x0001+BASE_CHUNK_PBA)
#define CHUNK_PBA_POOLGEOMS (0x0002+BASE_CHUNK_PBA)
#define CHUNK PBA GEOM01 (0x0003+BASE CHUNK PBA)
#define CHUNK PBA POOLANIMACIONES (0x0004+BASE CHUNK PBA)
#define CHUNK_PBA_ANIMACION01 (0x0005+BASE_CHUNK_PBA)
#define CHUNK PBA HUESOS (0x0006+BASE CHUNK PBA)
#define CHUNK PBA ANIMACIONNODO01 (0x0007+BASE CHUNK PBA)
#define CHUNK_PBA_POOLMATERIALES (0x0004)
#define CHUNK_PBA_MATERIAL01 (0x0005)
#define CHUNK_PBA_POOLTEXTURAS (0x0006)
#define CHUNK PBA TEXTURA01 (0x0007)
DWORD: Signature
DWORD: Version (some are 0x0000)
[Main chunk] CHUNK_PBA_MAIN
- BYTE[32]: Name
- [Bones chunk]
- [Textures chunk]
- [Geometry chunk]
- [Animation chunk]
[Bones chunk] CHUNK_PBA_HUESOS
- DWORD: Number of bones (can be zero)
```

- array of bones

BYTE[32]: szNodeName DWORD: idNodoPadre TTransform: transInitial

[Textures chunk]

DWORD: Number of textures - array of texture names

BYTE[32]

## [Geometry chunk]

- DWORD: Number of geometries
- array of mesh chunks

DWORD: Mesh type BYTE[32]: Mesh name

Vector3: Center as 3 floats

- FLOAT: Radius
- [Mesh data]

## Mesh type GEOMETRIA\_RIGIDA = 1:

- NumFaceGroups: DWORD
- array of face groups
  - DWORD: TextureId:
  - DWORD: AlphablendingFlags
  - DWORD: NumVertexIndices
  - Vertex Index array: WORD \* NumVertexIndices
  - If Signature > 1
  - DWORD: NumCivilisationTextures
  - array of civilisation texture names (each is BYTE[32])
- array of group vertices
  - DWORD: NumVertices
  - array of Vertices: TD3DVERTEXT1VC[NumVertices]

# Mesh type GEOMETRIA\_FLEXIBLE = 2 (unused in Praetorians)

- NumBlendPositions: DWORD
- Array of Blend Positions
- NumBlendNormals: DWORD
- Array of Blend Normals
- NumBlendVertices: DWORD
- Array of Blend Vertices
- NumFaceGroups: DWORD
- Array of Face Groups
- TextureId: DWORD
- NumVertexIndices: DWORD
- Array of Vertex Indices: WORD[]

#### Mesh type GEOMETRIA\_ANIMADA = 3:

- NumVertices: DWORD
- Array of Vertices: TD3DVERTEXT1VC []
- NumFaceGroups: DWORD
- Array of Face Groups
  - TextureId: DWORD
  - AlphablendingFlags: DWORD
  - NumFacegroupVertices
  - NumVertexIndices: DWORD
  - Array of Vertex Indices: WORD[]
  - If Signature > 1
  - DWORD: NumCivilisationTextures
  - array of civilisation texture names (each is BYTE[32])

#### Mesh type GEOMETRIA NODOS = 4:

- NumVertices: DWORD
- NumFaceGroups: DWORD
- Array of Face Groups
- TextureId: DWORD
- AlphablendingFlags: DWORD
- NumVertices: DWORD
- NumVertexIndices: DWORD
- Array of Vertex Indices: WORD[]
- NumVertexGroup: DWORD

- Array of VertexGroups
- name: char[32]
- NumVertices: DWORD
- Array of Vertices: TD3DVERTEXT1VC []
- If Signature > 1
- NumCivilisationTextures: DWORD
- Array of Civilisation Texture names (char[32])

### [Animation chunk]

NumSequences: DWORDArray of sequence chunks

## Sequence chunk:

- Name: char[32]
- Duration: DWORD
- NumSeqNodes: DWORD
- Array of SeqNode chunks
- NumAnimatedVertexSeq: DWORD
- Array of Animated Vertex chunks

## SeqNode chunk:

- NodeId: DWORD
- NumKeyframes: DWORD
- AnimType: DWORD
- Node Anim Keyframes

# Node Anim Keyframes (type ANIMACION\_ROT)

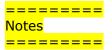
- Pos: Vector3 (3 floats)
- Array of KeyframeRot
- Time: DWORD
- Rotation: Quaternion (4 floats)

# Node Anim Keyframes (type ANIMACION\_ROTPOS)

- Array of KeyframeRotPos
- Time: DWORD
- Rotation: Quaternion (4 floats)
- Translation: Vector3 (3 floats)

## Animated Vertex chunks:

- NodeId: DWORD
- NumKeyframes: DWORD
- NumVertices: DWORD
- MinBox: TVector3 (3 floats)
- MaxBox: TVector3 (3 floats)
- Array of Keyframes
- Time: DWORD
- Vertex: Array of byteVector3 [NumVertices]
- 3 bytes, expand using the Min/Max coordinates



```
For GEOMETRIA ANIMADA type models:
- <u>NumFacegroupVertices</u> is used to compute the offset into the vertex array.
- For example:
   vertexOffset = 0;
   for each facegroup
      pVertexArray = vertexArray[vertexOffset]
      vertexOffset += NumFacegroupVertices
Node Anim Type:
ANIMACION_ROT = 1,
ANIMACION_ROTPOS = 2,
To convert a compressed (byte-sized) vertex into "real" Vector3:
Vector3 vertex = ((MaxBox - MinBox) / 255.0) * byteVector3 + MinBox;
TD3DVERTEXT1VC:
float x,y,z;
float nx, ny, nz;
dword uiRGBA;
float u,v;
AlphablendingFlags:
MATERIAL NONE = 0 \times 0000,
MATERIAL_ALPHA = 0x0001,
MATERIAL_ALPHATEST = 0x0002,
MATERIAL ALPHASOMBRA = 0 \times 0003,
struct TTransform
  Quaternion rotation; // 4 floats
  Vector3 translation; // 3 floats
Chunks are a header preceding some of the file's data pieces:
- Chunk ID: WORD
- Chunk length: DWORD
```