X3 Data Format Specification

Version 1.0e

24 September 2013

www.xojo3d.com

License

Document History

Date	Version	Author	Changes
19 Aug 2013	1.0	Alwyn Bester	First release.
5 Sep 2013	1.0b	Alwyn Bester	Added global color palette. Shortened names of polygon values. Minor grammar and spelling revisions.
17 Sep 2013	1.0c	Joe Astrahan	Normals as indexes.
		Alwyn Bester	Revised documentation with new normal indexes. Minor grammar fixes.
18 Sep 2013	1.0d	Alwyn Bester	Grammar fixes to polygon section.
24 Sep 2013	1.0e	Joe Astrahan	UV-coordinates as indexes.
		Alwyn Bester	Revised documentation with new UV indexes. Revised x3model schematic.

1. Introduction

The ***X3 Data Format*** specifies how to store and transmit 3D graphical assets, using the lightweight JSON data-interchange format. The primary goal of this specification is to provide a data format that is easy to use and implement in cross-platform environments.

This version of the X3 Data Format supports the following concepts:

Vertex
A point in 3D space described by an (X, Y, Z) coordinate.

Polygon
A collection of three or more vertices connected in such a way to form a closed path in an anti-clockwise direction.

Normal
The unit vector perpendicular to the surface of a polygon, indicating the direction that a polygon is facing.

RGBA Color Palette
A group of colors, with each color defined by red, green, blue and alpha components.

Texture
A 2D bitmap image that can be mapped (drawn) onto polygons.

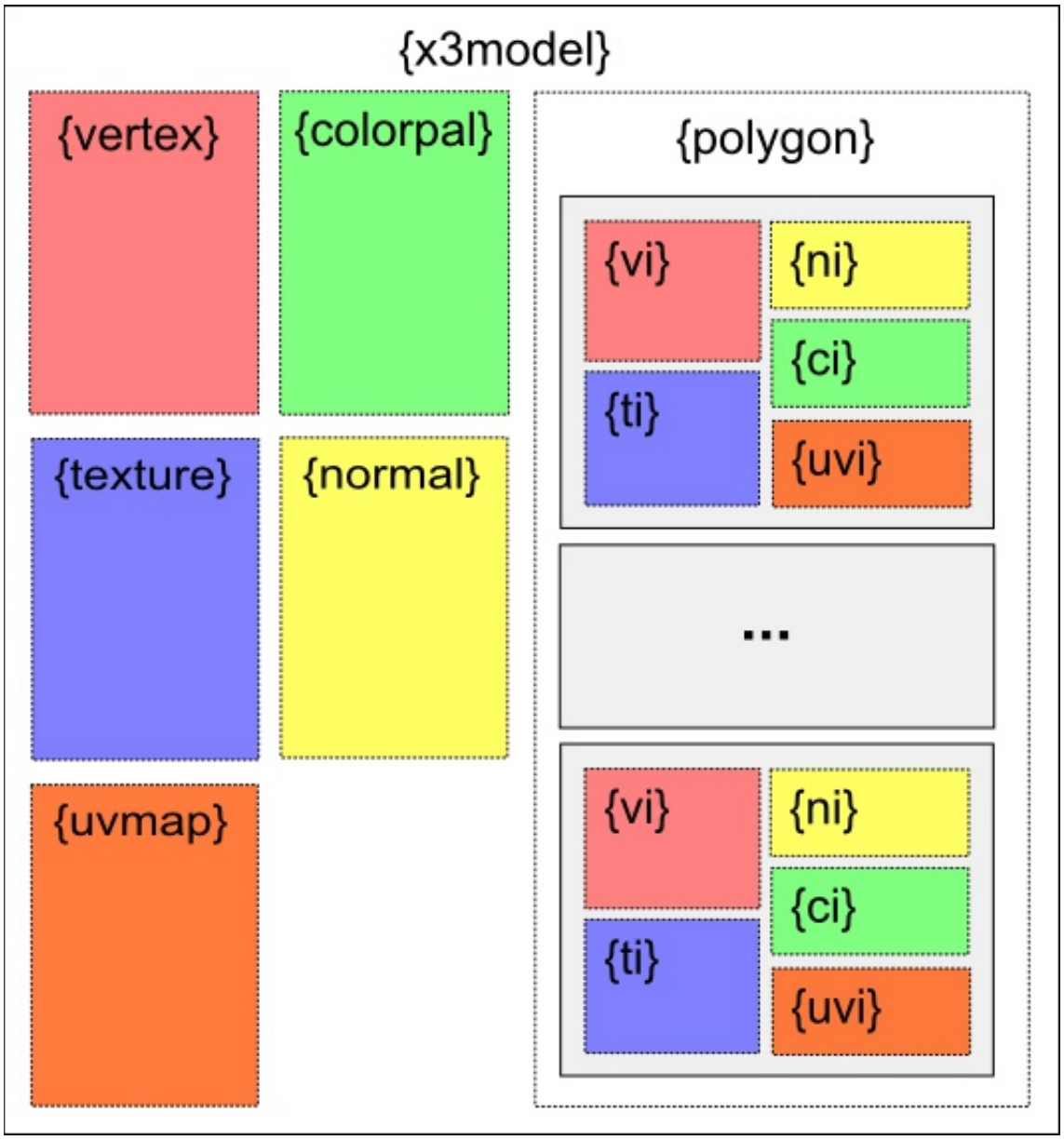
UV Map
A 2D coordinate system that defines how a texture is mapped to a polygon.

Model
An collection of polygons that form a complete 3D object.

2. JSON

The X3 Data Format is implemented with the JSON data-interchange format. The JSON format was chosen for its lightweight and cross-platform support.

The following schematic illustrates the high-level structure of the X3 Data Format:



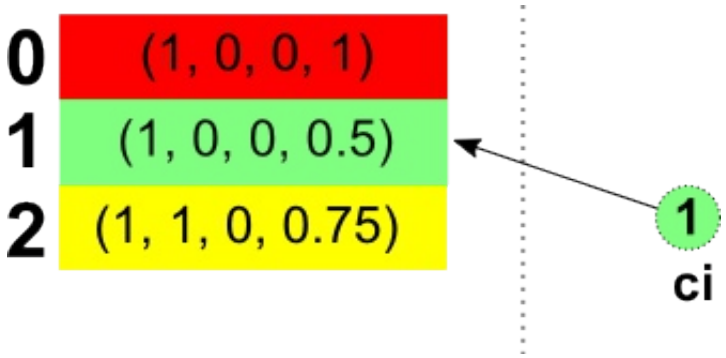
The following sections describe each X3 data element in detail.

ci

Description: ***ci*** is an integer index that points to a color stored in the global color palette of the model.

Child Values: None

Example: `"ci":1`



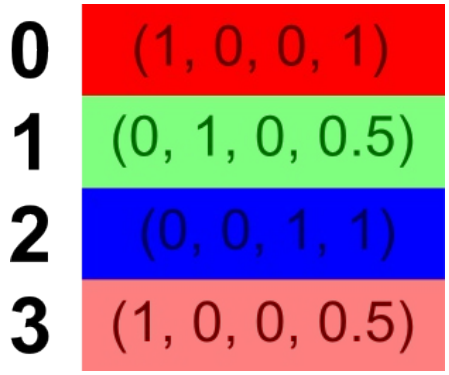
colorpal

Description: **colorpal** is an array of numerical values that define all the colors in the color palette of a model. Each color has red, green, blue and alpha values that are stored sequentially in the array.

Child Values: None

Example:

```
"colorpal":[1, 0, 0, 1,
            0, 1, 0, 0.5,
            0, 0, 1, 1,
            1, 0, 0, 0.5]
```

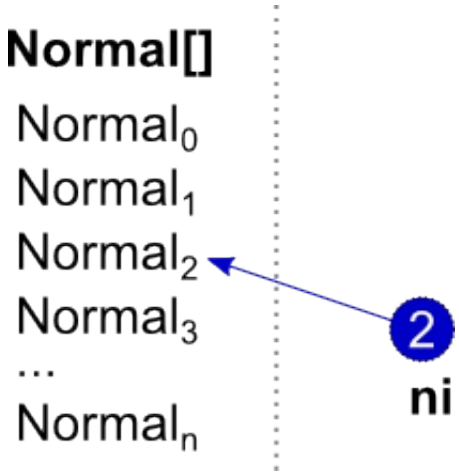


ni

Description: ***ni*** is an integer index that points to a normal vector stored in the global normal array of the model.

Child Values: None

Example: `"ni":2`



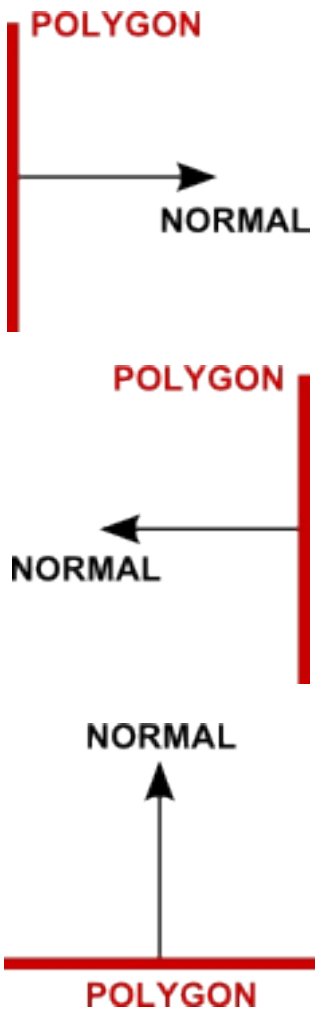
normal

Description: ***normal*** is an array of numerical values that define normal vectors in 3D space. Each normal vector has an (X, Y, Z) coordinate that is stored sequentially in the array.

Child Values: None

Example 1:

```
"normal":[1, 0, 0,
          -1, 0, 0,
          0, 1, 0]
```



polygon

Description: ***polygon*** is a global array that stores all the polygons used by the model. Each polygon in the array contains all the information needed to render the polygon, e.g. vertex information, color information, texture information and normal vector.

Child Values: ***ci***
An index value that points to an RGBA color in the model's global color palette. ci is ignored when ti is greater than or equal to 0.

***n*i**
An integer index value that points to a normal vector in the model's global normal vector array.

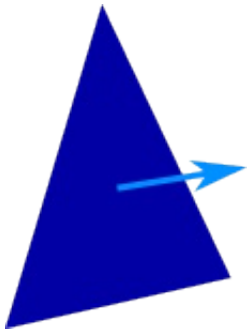
ti
An integer index value that points to a texture in the model's global texture array. When tindex is greater than or equal to 0, the color of the polygon is ignored.

uvi
An array of indexex that point to uv-coordinates stored in the model's global uvmap array.

vi
An array of three or more index values that point to vertices stored in the model's global vertex array.

Example:

```
"colorpal":[0, 0, 0.7, 1],
"normal":[0, 0, 1],
"vertex":[0, 1, 0, -1, -1, 0, 1, -1, 0],
"polygon":[{
     "vi":[0, 1, 2],
     "ci":0,
     "ni":0}]
```



texture

Description: ***texture*** is a global array that stores all the textures used by the model. Each texture is stored as a PNG bitmap image encoded as a BASE64 string.

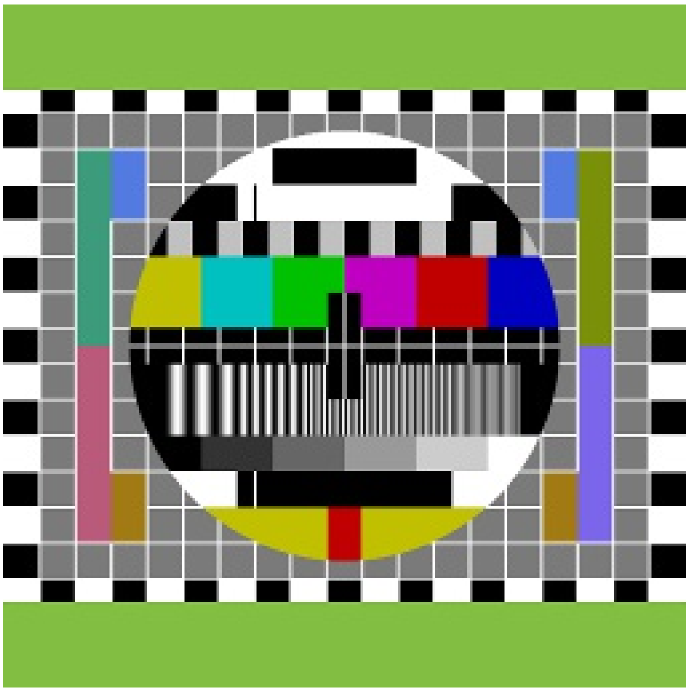
Child Values: None

Example:

```
"texture":["iVBORw0KGgoAAAANSUhEUgAAAgAAAAIACAYAAAD0eN
T6AAAABmJLR0QA/wD/AP+gvaeTAAAACXBIWXMAAAsSAAALEgHS3X78
AAAAB3RJTUUH3QUbCxgAo8KNBwAAIABJREFUeNrs3Xl4FeWhP/DvLG
fOkuSE7AkhhD0kgCBhiSy1IiK4AMUFqFCtYnuvULda29r7e37Pfbrp

...

AAAIAAAAgAAAAAAIAAAAgAAAAAAIAAAAgAAAAAAIAAAAEAAAAQAAAA
AAEAAAAQAAAAAAEAAAAQAAAAAAEAAAAQAAAAAAEAAAAQAAAAAAEAAA
AQAAAAAAEAAAAQAAAAAAEAAAAQAAAAAAEAAAAQAAAAIAAAAAACAAAA
IAAAAAACAAAAIAAAAAACAAAAIAAAAAACAAAAIAAAAAACAAAAIAAAAA
ACAAAAIAAAAAACAAAAIAAAAAACAAAAIAAAAAAAQAAABAAAAAAAQAAA
BAAAAAAAQAAABAAAAAAAQAAABAAAAAAAQAAABAAAAAAAQAAABAAAAD
ATfB/wMdbXBQFr9oAAAAASUVORK5CYII="]
```

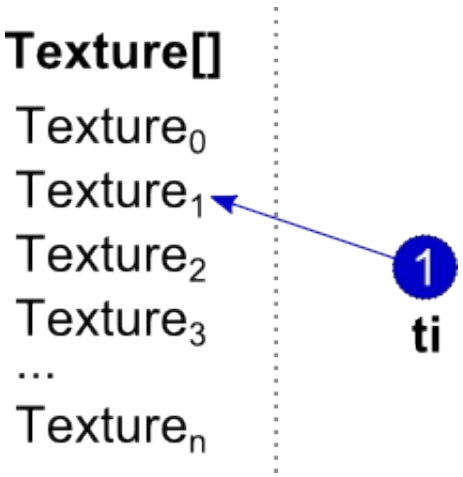


ti

Description: ***ti*** is an integer index that points to a texture stored in the global texture array of the model. The uvi index array defines how the texture is mapped to the polygon.

Child Values: None

Example: `"ti":1`



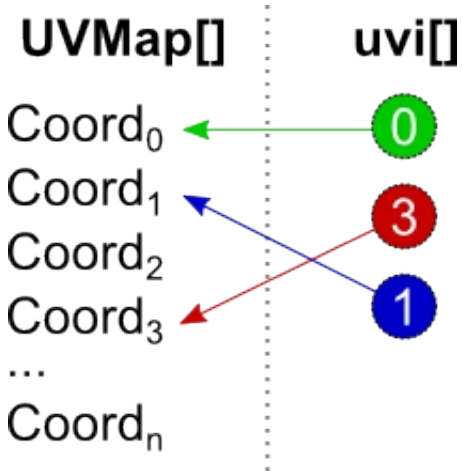
uvi

Description: ***uvi*** is an array of integer values that point to uv-coordinates stored in the global uvmap array of the model.

IMPORTANT: The number of elements in uvi need need to exactly match the number of vertices of the polygon.

Child Values: None

Example: `"uvi":[0, 3, 1]`

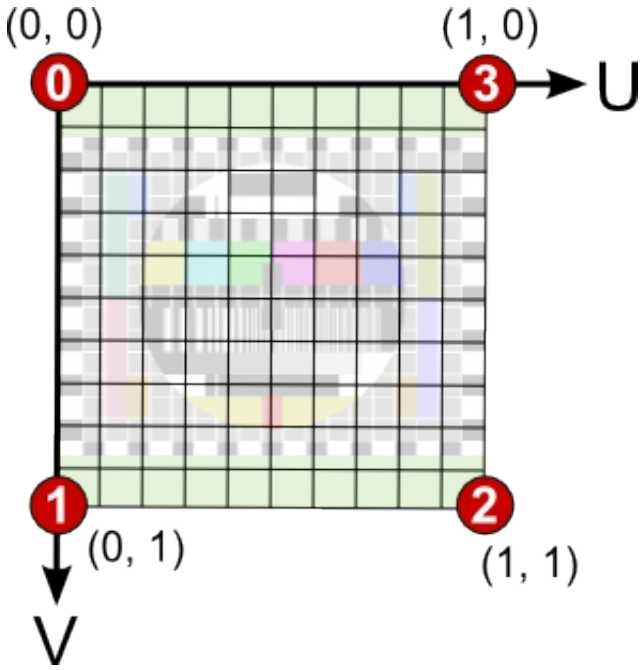


uvmap

Description: **uvmap** is an array of (U, V) coordinates that defines how textures are mapped to polygons.

Child Values: None

Example: `"uvmap":[0, 0, 0, 1, 1, 1, 1, 0]`



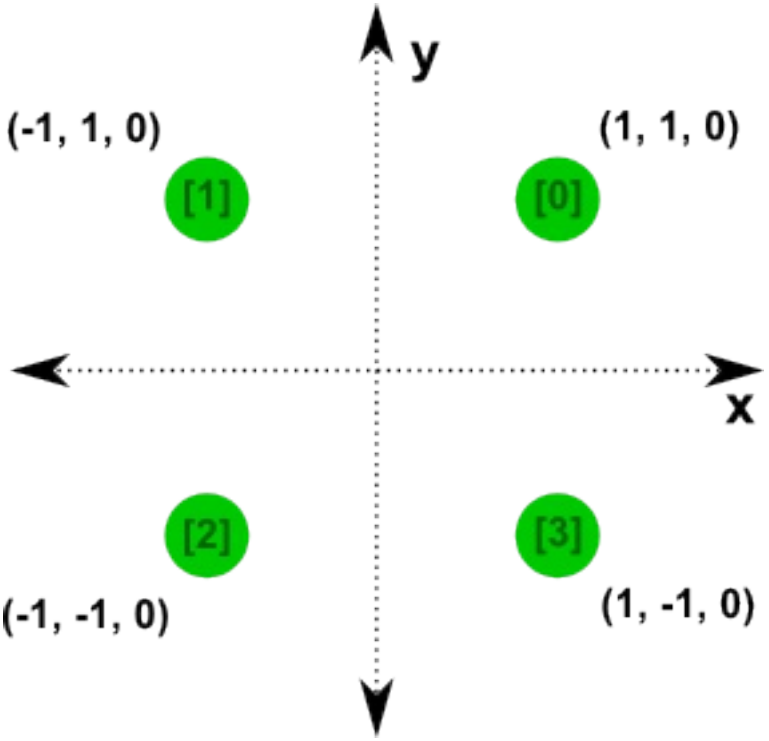
vertex

Description: ***vertex*** is an array of numerical values that define points in 3D space. Each point has an (X, Y, Z) coordinate that is stored sequentially in the array.

Child Values: None

Example:

```
"vertex":[
     1, 1, 0, -1, 1, 0, -1, -1, 0, 1, -1, 0
]
```

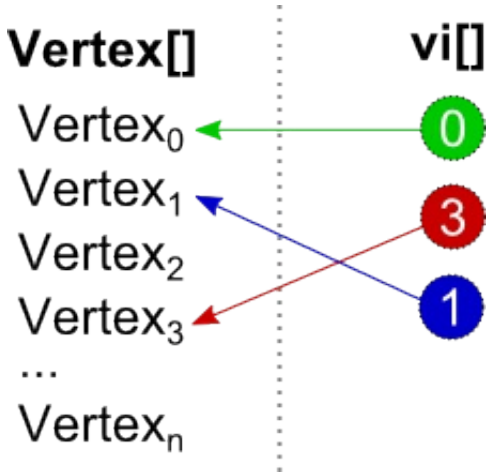


vi

Description: ***vi*** is an array of integer values that point to vertices stored in the global vertex array of the model.

Child Values: None

Example: `"vi":[0, 3, 1]`



x3model

Description: ***x3model*** defines a 3D object, together will all the relevant information needed to render the object, e.g. vertices, colors, normals and textures.

Child Values:

colorpal
An array that stores all the RGBA colors used by the model.

normal
An array that stores all the normal vectors of the model.

polygon
An array that stores all the polygons that make up the model.

texture
An array that stores all the textures used by the model.

uvmap
An array that stores all the uv-coordinates used during texture mapping.

vertex
An array that stores all the vertices of the model.

Example:

```
"x3model":{
     "colorpal":[0.7, 0, 0, 1, 0, 0.7, 0, 1],
     "vertex":[
          1, 1, 0, -1, 1, 0, -1, -1, 0, 1, -1, 0
     ],
     "polygon":[
     {
          "vi":[0, 1, 3],
          "ci":0
     },
     {
          "vi":[1, 2, 3],
          "ci":1
     }
     ]
}
```