

# OPTIMIZED NESTING ENVIRONMENT (ONE)

The optimized nesting environment (ONE) is a new Ilumbra led format for storing all volumes, textures and scene parameters for the correct display of each 3D volumetric model. The format stores a list of volumes (region in the scene) and their associated textures. The texture(s) used by a volume is determined by the volume's TEXTURE\_ID\_X parameter (see below).

## Object Definitions

**Scene:** Contains all the textures, volumes and information needed to render a model.

**Texture:** Non-zero voxels covering a volume in space.

**Volume:** A distinct region in the scene.

**Voxel:** An rgba value at a given location in space.

A ONE file is ordered with the data FIRST and then the header. This allows for quick header writing without having to re-read the large textures.

Each ONE object has a list of parameters (described below).

## File Format (.ONE)

The following describes the format of the ONE file. Each entry in **bold** indicates the type of variable saved/read. The **string** variable is encoded using a modified-utf-8 format. The first two bytes of the **string** variable represents the number of bytes in the following string, which is encoded in the modified UTF-8 format. The data is stored in Big Endian format, with an **int** being 4 bytes, a **float** 4 bytes, and a **long** 8 bytes.

In the case of voxels, the data can either be written as **float** or **byte** depending on the texture TYPE parameter. If the texture TYPE parameter is RGBA\_FLOAT the voxel data will be **float**, and if the texture TYPE parameter is RGBA\_BYTEx the voxel data will be **byte**.

The parameter **string** read for each ONE object consists of all parameters concatenated by the "!" "@" string. Each parameter consists of a KEY and VALUE string, separated by a ':' character. As an example, the parameter string from the scene object may look like:

```
"TYPE:NESTED!@EXPOSURE:15.0!@EMISSION:1.0!@OPACITY:600.0"
```

In this example, there are 4 parameters: TYPE:NESTED, EXPOSURE:15, EMISSION:1.0 and OPACITY: 600.

The steps to read the ONE file should be similar to the following:

1. Seek to the LAST long value of the file. This will indicate the length of the header.
  - a. filePos = fileSize - longSize
  - b. Seek(filePos)
  - c. headerLength = readLong()
2. Seek to the start of the header.
  - a. headerStartPos = fileSize - headerLength - longSize;
  - b. Seek(headerStartPos)
3. Read the Header
4. Seek to the beginning of the file.
5. Read the texture data. The number of textures to read will be available from the number of textures read from the header.

## File Structure

### Data

- For each Texture
  - **long** : Texture ID
  - **int** : Number of Voxels
  - For each Voxel:
    - **int** : x index of voxel
    - **int** : y index of voxel
    - **int** : z index of voxel
    - **byte/float** : r value
    - **byte/float** : g value
    - **byte/float** : b value
    - **byte/float** : a value

### Header

- **int**: ONE File ID = 102380. Used to identify ONE files.
- **int**: ONE version
- Scene
  - **long** : Scene ID
  - **string** : Scene name
  - **string** : Parameter string (see above)
- Volumes
  - **int** : Number of Volumes in the scene
  - For each Volume
    - **long** : Volume ID
    - **string** : Volume name
    - **string** : Parameter string (see above)
- Textures
  - **int** : Number of Textures in the scene
  - For each Texture
    - **long** : Texture ID
    - **string**: Texture name
    - **string**: Parameter string (see above)
- **long** : Header length in bytes

## Common Parameters

Below is a list of common parameters to describe how to orient the volumes and render the scene. There is no limit to how many parameters can be included.

### **ONE SCENE**

Parameter Key	Description
TYPE	The type of model. e.g. SINGLE, NESTED
EXPOSURE	The factor by which to multiply the overall rendered pixel values.
EMISSION	The factor by which to multiply the rgb values of each voxel.
OPACITY	The factor by which to multiply the 'a' values of each voxel.
ROT_X	The x-rotation (deg) of the model to achieve the 'from earth' viewpoint.
ROT_Y	The y-rotation (deg) of the model to achieve the 'from earth' viewpoint.
ROT_Z	The z-rotation (deg) of the model to achieve the 'from earth' viewpoint.

## ONE VOLUME

Parameter Key	Description
TEXTURE_ID_X	The ID (s) of the texture (s) this volume will render.
ORDER	The order of the volume in the scene. Smaller order volumes are 'on top' of higher order volumes.
BLEND	How much to blend the volume with the one 'above' it (0-1)
REPLACE	Should this volume replace the volumes above it? (true / false)
EMISSION	The factor by which to multiply the rgb values of each voxel in the volume.
OPACITY	The factor by which to multiply the a values of each voxel in the volume.
SCALE	The size of the volume with respect to the entire scene.
OFFSET_X	The x-position of the volume with respect to the scene. (-1 to 1)
OFFSET_Y	The y-position of the volume with respect to the scene. (-1 to 1)
OFFSET_Z	The z-position of the volume with respect to the scene. (-1 to 1)
ROT_X	The x-rotation (deg) of the volume within the scene.
ROT_Y	The y-rotation (deg) of the volume within the scene.
ROT_Z	The z-rotation (deg) of the volume within the scene.

## ONE TEXTURE

Parameter Key	Description
TYPE	The type of data represented in this texture. Values may be one of RGBA_FLOAT or RGBA_BYTE
WIDTH	The width of the texture cube.
HEIGHT	The height of the texture cube.
DEPTH	The depth of the texture cube.
CONTENT_TYPE	The type of content stored in the texture. One of VOLUME, PROCEDURAL, DATA
MAX_GREY	The maximum grey value in the texture. The grey value is computed as $(r + g + b) / 3$ ;
MAX_A	The maximum 'a' value in the texture.
MIN_X	Index of the first voxel that contains non-zero colors in the x direction.
MIN_Y	Index of the first voxel that contains non-zero colors in the y direction.
MIN_Z	Index of the first voxel that contains non-zero colors in the z direction.
MAX_X	Index of the last voxel that contains non-zero colors in the x direction.
MAX_Y	Index of the last voxel that contains non-zero colors in the y direction.
MAX_Z	Index of the last voxel that contains non-zero colors in the z direction.
JSCALE	Scaling that should be applied to the rgb values.
KSCALE	Scaling that should be applied to the 'a' values.