Requirement Specification for openPlan

# Intro

LinkSlicerBlender provides a framework for communication between the open source 3D Modelling software Blender and the open source 3D medical imaging and analysis package 3DSlicer. The goal of LSB is to allow leverage of the core features of both softwares to be used in harmony. This document describes the performance requirements of the software. In short, what the software should and should not do.

# Transfer of Objects

The transfer of objects directly from one software to the other allows for streamlining workflows by circumventing slow and laborious manual export/save/import between the two softwares by the user.

-Mesh data objects in Blender can be transferred to Slicer

Triangles and quads only

Mesh data with Ngons will be triangulated or rejected

-Mesh data objects in Slicer can be transferred to Blender

Triangles and quads only.

-Non Mesh objects at this time cannot be transferred.

-Non mesh objects will be prevented from syncing from Slicer to Blender

-Non mesh objects will be prevented from syncing from Blender to Slicer

-Automatic updating of mesh deformations is not supported. The object will need to be re-sent

-Re-sending an object from Blender to Slicer that has previously been synced will(?) overwrite the object mesh data in 3DSlicer

-Will there bea warning?  
  
-Re-sending an object from Slicer to Blender that has previously been synced will(?) overwrite the object mesh data in Blender

-Will there bea warning?

-Objects sent from Slicer to Blender will have the same name in both scene management outliners

-Objects sent from Blender to Slicer will have the same name in both scene management outliners

# 

# Transform Management

Once an object exists in both 3DSlicer and Blender, it is important that their spatial relationship in the 3d world be identical.  
  
  
-The world space coordinate systems are the same in Blender and 3DSlicer. Both are right handed, orthogonal coordinate systems.

-Linked objects will maintain the same transform in both softwares

Linked objects which are manipulated in Blender will have the following equivalent manipulations synchronized in 3DSlicer.

* Location
* Rotation
* Scale

Linked objects which are manipulated in 3DSlicer will have the following equivalent manipulations synchronized in Blender.

* Location
* Rotation
* Scale

Deformations in the mesh object (sculpting, booleans etc) will not be reflected unless the object is retransfered (See Above)

LSB PRO: Patrick: “I can also advise a way to detect mesh changes to try and alert out of sync meshes on both ends. This is likely to go in master, later.”

# Color Management

Color management is a good way to help the user identify different objects rapidly in 3DSpace. Ensuring that the objects are visualized in as similar a way as possible

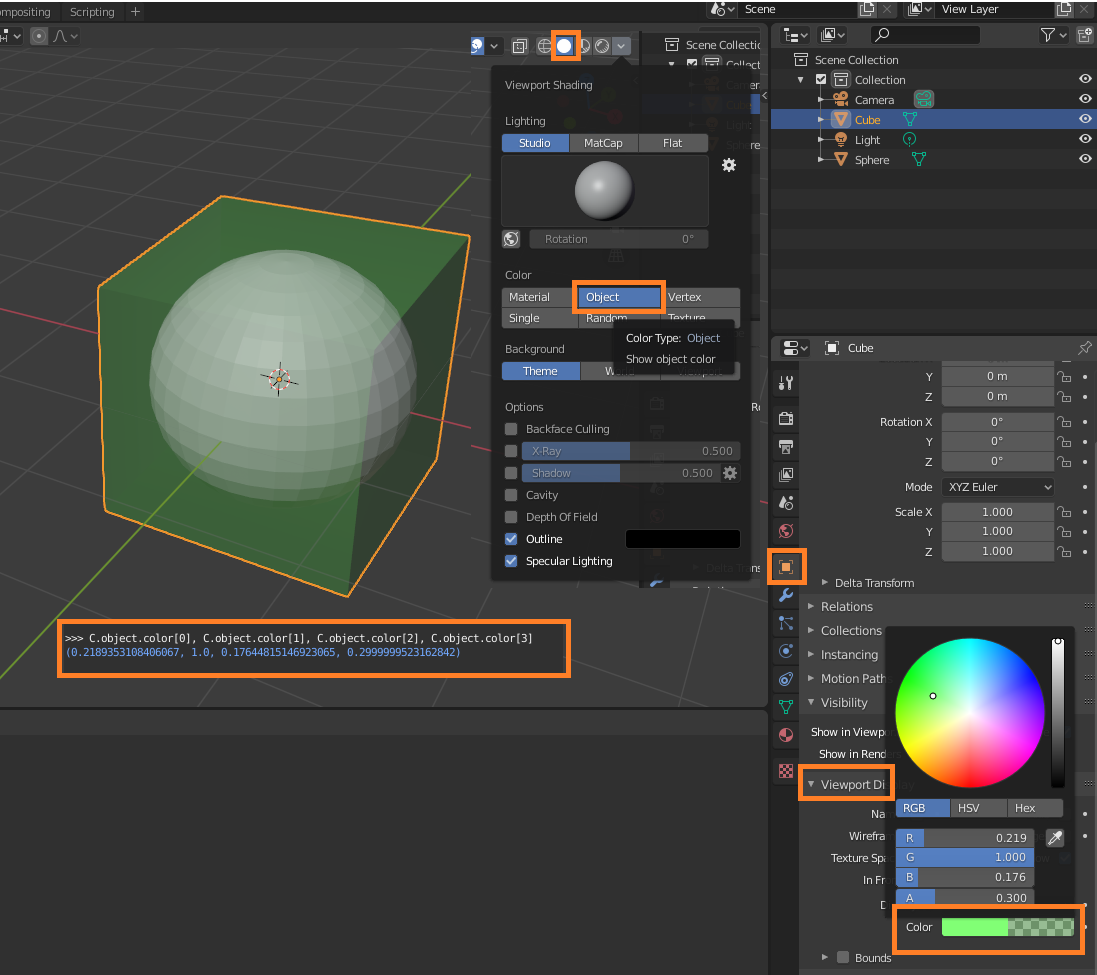
-Simple single color syncing between Blender and Slicer is possible.

-The Viewport Display Color which displays on the object surface in Blender’s **Solid Viewport Mode** as rendered by the Workbench Render engine will have the same RGB(A) value in 3D Slicer’s vtkMRMLViewNode and vtkMRMLSliceNode UI widgets color, and vice versa.  
-Syncing the rendering/visualization of objects with **Node Base Materials, Cycles Materials, Transparency, etc is not supported**.

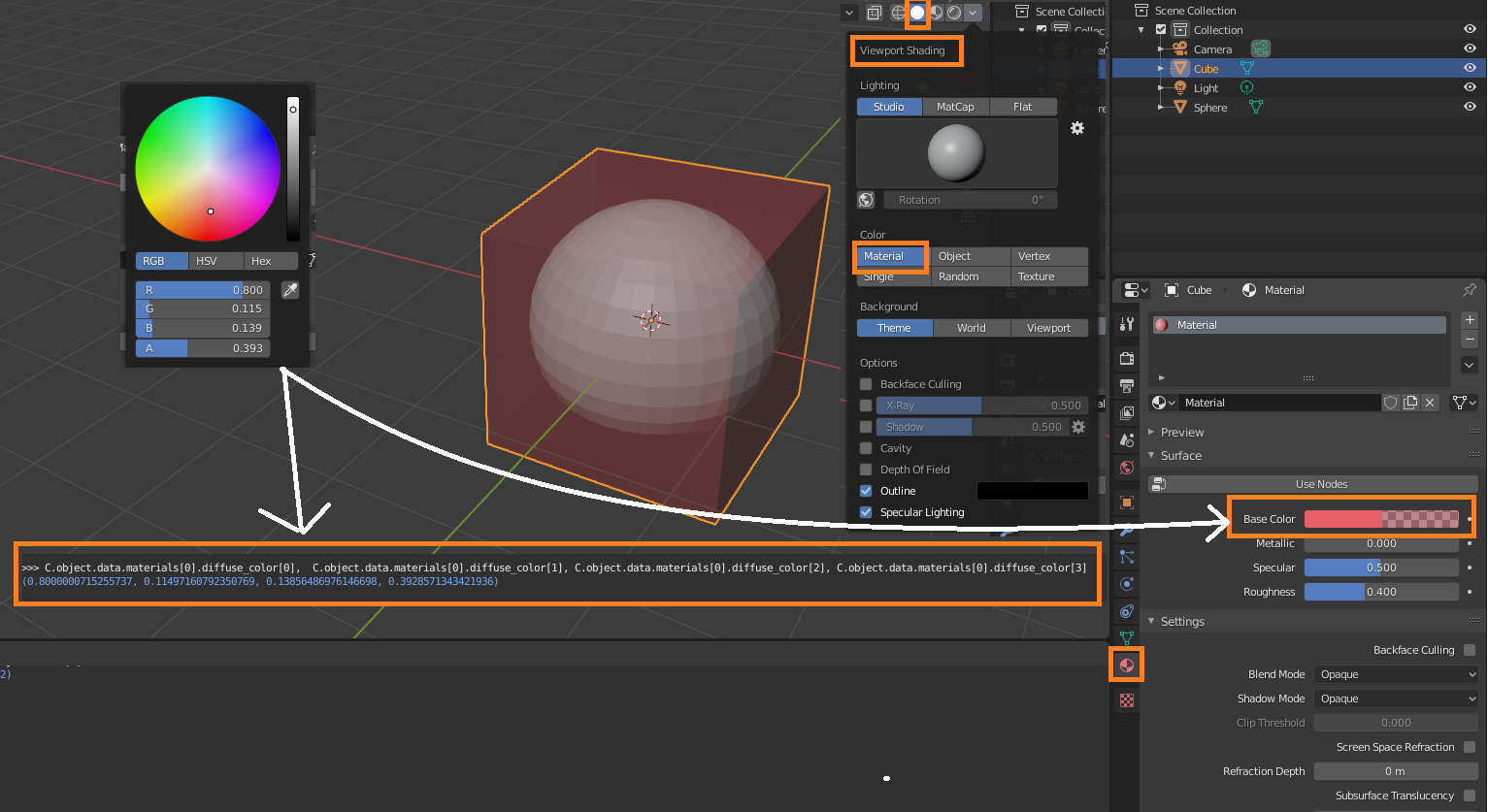
-A context aware warning will be displayed in the LSB panel if the current Blender viewport settings are not compatible with visualization equivalents  
-LSB will not prevent the use of those visualizations in Blender, however the display in Slicer will not be equivalent.

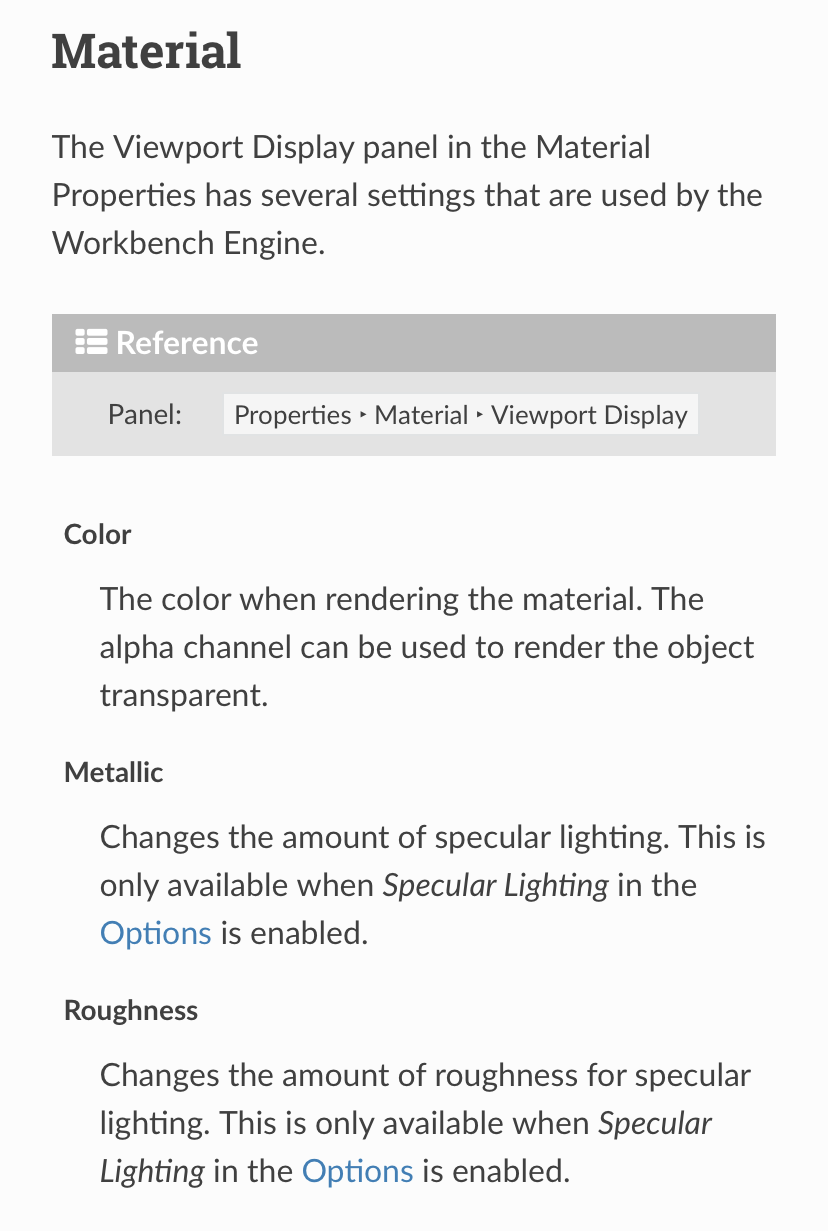
LSB PRO: alpha transparency and selective show of planes in 3d slicer widgets (important when bridging LSB Pro w/ slicerPano Pro as the planes are used for visualization in 3D slicer and should not show in all views)

In terms of color management, where the color comes from. We can pick or detect



OR





-User changing the color of the object in Blender, updates the color of the object in Slicer

-LSB PRO: User changing the color of the object in Slicer will update the color of the object in Blender

Design Specification for openPlan

# Intro

This section of the document discusses how the requirements in the requirement specification (above) are to be achieved with some discussion of code.

* Asyncsock.py
  + Provides a basic TCP/IP communication framework
  + Provides 3D Slicer class (dependent on Qt) to establish TCP/IP client connection to a host, maintain that connection until closed, send and receive byte code data.
  + Provides Blender class (depends on native Python3 thread management and async classes) to create TCP/IP server host which accepts a single client connection from one 3D slicer (ie. multiple 3D slicer connections are not possible), maintains that connection until closed, sends and receives byte code data.
* Bidirectional object polydata copy
  + **Transfer rate based on object size (in RAM). Models with high poly count will copy slower.**
  + There is **no current UI implementation to show a progress bar** of this process as it occurs in the background.
  + **Model linking and transform synchronization is dependent on object name, e.g. there is a model of a bunny in 3D Slicer and a model of a monkey in Blender. Both models are named “my\_model” and even though geometrically they are different, linkSlicerBlender identifies them as the same because they share the same object name.**
  + While transfer is occurring in the background, the Blender and 3D Slicer UIs remains responsive.
* Linking: Object matrix transform and RGB viewport color
  + The data link is unidirectional: Blender object -> 3D Slicer object
  + An object is considered “linked” when it is added to the “slicer\_link” collection within Blender.
  + When the object(s) is in the “slicer\_link” collection, any transformation (move, rotate, scale) and viewport RGB color applied to this object(s) is transferred to 3D Slicer via the established TCP/IP connection.
  + 3D Slicer received the transform and RGB color data and applies it to the object(s)
  + Feature request by Michael: Opacity control link Blender -> 3D Slicer
    - <https://discourse.slicer.org/t/model-opacity-change-in-single-view/4707/2>
* Blender Server
  + Host: (default 127.0.0.1 or localhost)
    - The server’s IP binding. Although theoretically this can be done over any network (local or internet), the addon is intended for localhost virtual adapter use only.
  + Port: (default 5959)
    - The server’s port binding. Port number is arbitrary and any can be used as long as there are no other services utilizing the port.
  + “Server” Button
    - Changes to “Slicer Link Stop” when the server instance is created and listening to the configured host:port. Throws an exception if a server/client instance is utilize the host:port
    - “Link Object(s)” Button
      * Operates on a single or multiple object(s) selected within the Blender scene.
      * If an object is present in 3D Slicer by the same name but not linked then a link is established by placing the object in “slicer\_link” collection.
      * If an object is not present in 3D Slicer by the same name then the polydata of the object is copied from blender into 3D Slicer, and linked.
    - “Unlink Object(s)” Button
      * Operates on a single or multiple object(s) selected within the Blender scene.
      * If object(s) is linked, ie. name is present in “slicer\_link” collection, then it is removed from “slicer\_link” collection and therefore unlinked.
    - “Delete Object(s)” Button and Drop-down
      * Operates on a single or multiple object(s) selected within the Blender scene.
      * Delete -> Blender (default)
        + Unlinks object(s)
        + Deletes object(s) from Blender scene.
      * Delete -> 3D Slicer
        + Unlinks object(s)
        + Deletes object(s) from 3D Slicer scene.
      * Delete -> Both
        + Unlinks object(s)
        + Deletes object(s) from both Blender and 3D Slicer scenes
  + **“Client” Button - to be removed**
* **LSB PRO: On save, hash calc mesh from both sides to detect poly data changes, resolve changes, then save**
* 3D Slicer Client
  + Host: (default 127.0.0.1 or localhost)
    - The host server IP address to connect to. Although theoretically this can be done over any network (local or internet), the addon is intended for localhost virtual adapter use only.
  + Port: (default 5959)
    - The host server port to connect to. Port number is arbitrary and any can be used as long as there are no other services utilizing the port.
  + “Connect” Button
    - Establishes the connection to the Blender server host based on the provided host and port parameters.
    - The button changes label text to “Stop.” This only indicates that a connection attempt is made. If no connection is made, press the “Stop” button again so it turns to “Connect,” then press the “Connect” button. This effectively resets the connection to the host server.
  + “Add Model” Button
    - If a polydata model is present within the 3D Slicer scene, this button allows the model to be linked with Blender.
    - A new drop-down menu appears allowing the user to select a model to be linked. If no model is selected, pressing the “Add Model” button does not add another drop-down menu. An additional model can be added once the drop-down menu slot has been used.
      * This prevents the user from adding infinite number of empty model drop-down selection panels
    - If there are multiple linked models, changing any one of them to “None” causes the model to be unlinked from Blender and the drop-down menu instances are discarded from the UI.
      * This allows for object link/unlink management to occur from within 3D Slicer UI.
    - Adding a model logic:
      * With the above statement in mind, if model exists in Blender scene, then a transform matrix link is established
      * If the model does not exist in the Blender scene then 3D Slicer decimates (CPU intensive) the poly data (based on coded parameters - subject to change and adjustment) and copies to Blender via TCP/IP. Then the transform matrix link is established.

slicerPano Pro (sP Pro)

* XML UI layout providing 5 views (overrides current 3D Slicer UI layout when add-on is launched)
  + Anatomical Transverse Plane
    - Default to 3D slicer
  + Path Transverse (not anatomic transverse plane)
    - Follows the selected defined open curve path and renders the DICOM voxels and 3D model slice cross-section that is normal in position to the curve with tangent orientation
  + Tangential
    - Same as Path Transverse slice widget but with the ability to rotate about the axial line (normal to transverse anatomical plane) at the current path position.
  + Free View
    - Same as tangential but it has an additional degree of rotational freedom about the selected curve path’s tangent line.
* View widgets are attached to add-on’s slider’s within the panel UI for control.
* Add-on UI Panel
  + “Path” Collapsible Button
    - Input Volume Drop-down / Select Volume Button
      * When multiple DICOM volumes are loaded, the user can choose which is the active volume to be displayed in the views.
    - Input Curve Drown-down / Select/Update Curve Button
      * The user can select which open curve is used for the path projection.
  + “Slice Controls” Collapsible Button
    - Transverse slider attached to Path Transverse view - value range calculated based on open curve total length
    - Tangential slider attached to the Tangential view - value range 0 - 360 degrees
  + “Free View Controls” Collapsible Button
    - Tangential Angle slider attached to Free View slice widget and controls the rotation as described in “Tangential” XML UI element
    - Axial Angle slider attached to Free View slice widget and controls the rotation about the open curve’s tangent line at location origin

PANTOMOGRAPHIC RECONSTRUCTION NOTES FROM 3D SLICER DISCOURSE

<https://discourse.slicer.org/t/how-to-implement-cpr-curved-planar-reconstruction-from-centerline/9456/16>

<https://discourse.slicer.org/t/merge-ct-volume-node-with-a-model-node-in-python/9457/5>

