

## LUATrayMesh

[Desktop Automation]

The `LUATrayMesh` object contains the data of parts in a fabbproject's buildroom, listed in the project tree, i.e. an instance of `LUAMesh` (see [Mesh \(?guid=NETF-LUA-MESHOBJ\)](#)) and its position, scale and rotation. Traditionally, these are referred to as "mesh", but they are more than purely a triangle net and can also be tessellated parametric, or CAD parts. To access the triangle net and use LUAMesh properties use the `mesh` property.

- [Properties](#)
- [Methods](#)
  - [Packing options](#)
    - [Read/Write parameters](#)
    - [Read-only parameters](#)
    - [Example: Reset packing options for all parts](#)

### Properties

Property	Read/write	Type	Description
area	read	Number	Get the area of the traymesh (including transformation, in mm <sup>2</sup> )
color	read/write	Number	Set or get the display color. Note: When you set a new color, make sure to run <code>application:triggerdesktopevent('updateparts')</code> some time after, otherwise the display color change remains unreflected in the actual UI on screen.
fileorigin	read	String	If the part was imported from a file, or derived from a part that was imported from a file, this property contains the full file system path.
id	read	Number	Internal ID of the traymesh. This is not the part number as displayed in the small box next to the part name in the project tree.
isclosed	read	Boolean	Whether the triangle net of this traymesh is closed
matrix	read	LUAMatrix4f	Get the transformation matrix of the traymesh
mesh	read	LUAMesh	Get the triangle net of the traymesh. Returns a new LUAMesh object.
hassupport	read	boolean	Returns true if the traymesh has attached support, false otherwise
name	Read/write	String	Name of the traymesh as it appears in the project tree
outbox	read	Outbox	Bounding box object
outboxbasearea	read	Number	Get the bounding box base area of the traymesh (including transformation, in mm <sup>2</sup> )
outboxheight	read	Number	Get the bounding box height of the traymesh (including transformation, in mm)
outboxvolume	read	Number	Get the bounding box volume of the traymesh (including transformation, in mm <sup>3</sup> )
parent	Read/write	LUAMeshGroup	Get or set the group the traymesh belongs to
selected	read/write	Boolean	Sets or returns selection state of the traymesh
support	read	LUAMesh	Get the triangle net of the attached support for that traymesh. Returns a new LUAMesh object. Operation fails if the traymesh does not have support.
visible	read/write	Boolean	Visibility of the traymesh in the 3D view. Transparent display counts as visible, or true, but cannot be set through Lua, only read.
uuid	read	String	UUID of the traymesh
selected	read/write	Boolean	Sets or returns selection state of the traymesh
volume	read	Number	Get the volume of the traymesh (including transformation, in mm <sup>3</sup> )

[Back to top](#)

Methods

Name	Syntax	Description
assignsupport	mainmesh:assignsupport(supportmesh:LUAMesh; relativecoordinates:Boolean)	Assigns a triangle net as a support to a traymesh. When relativecoordinates is true, supportmesh is applied relative to traymesh. When supportmesh and redativecoordinates are omitted, any existing support is detached from traymesh.
createsupportedmesh	mesh:createsupportedmesh( mergepart:Boolean, mergeopensupport:Boolean, mergeclosedsupport:Boolean, openthickening:number)	Returns a supported traymesh. The merge... parameters determine which mesh type is included. The thickening parameter extrudes open support into solid, closed mesh.
getpackingoption	Option = traymesh:getpackingoption(OptionIdentifier: String)	Get a packing option of the traymesh
getuuid	Uuid = mesh:getuuid()	Get the UUID of a traymesh
rotate	traymesh:rotate(axis_x, axis_y, axis_z, angle, cen_x, cen_y, cen_z: Number)	Rotate a traymesh by an angle around a selected axis around a given center against the platform coordinates. The axis is "selected" by a positive, non-zero number. Rotate angle is in radians. The center is optional. If no center is given, the center of the traymesh part's bounding box is used.
saveto3ds	traymesh:saveto3ds(filename:String)	Export the traymesh including all transformations to 3DS
saveto3mf	traymesh:saveto3mf(filename:String)	Export the traymesh including all transformations to 3MF
savetoamf	traymesh:savetoamf(filename:String)	Export the traymesh including all transformations to AMF
savetoasciistl	traymesh:savetoasciistl(filename: String)	Export the traymesh including all transformations to ASCII STL
savetogts	traymesh:savetogts(filename:String)	Export the traymesh including all transformations to GTS
savetoncm	traymesh:savetoncm(filename:String)	Export the traymesh including all transformations to NCM
savetoobj	traymesh:savetoobj(filename:String)	Export the traymesh including all transformations to OBJ
savetoply	traymesh:savetoply(filename:String)	Export the mestraymeshh including all transformations to PLY
savetostl	traymesh:savetostl(filename:String)	Export the traymesh including all transformations to STL
savetovrml	traymesh:savetovrml(filename:String)	Export the traymesh including all transformations to VRML
savetox3d	traymesh:savetox3d(filename:String)	Export the traymesh including all transformations to X3D
savetozpr	traymesh:savetozpr(filename:String)	Export the traymesh including all transformations to ZPR
scale	traymesh:scale(x, y, z:Number)	Scale a traymesh
setmatrix	traymesh:setmatrix(matrix:LUAMatrix4)	Overwrite the transformation matrix of the traymesh with a new one
setpackingoption	traymesh:setpackingoption(option:String; value:String)	Add a packing option to the traymesh
shellasmesh	Newmesh = mesh:shellasmesh(shellnumber:number)	Extract a shell as new LUAMesh object. Same as <code>LUAMeshObject.shellasmesh</code> but takes the matrix in the fabbproject into account. See <code>LUATrayMesh.mesh.shellcount</code> for the number of meshes.
translate	traymesh:translate(x, y, z: Number)	Translate a traymesh

[Back to top](#)

Packing options

Read/Write parameters

```
- priority:number (, 1-10)
- restriction:string ("locked", "norestriction")
- rotate:string ("arbitrary", "zaxis", "forbidden")
```

These options must be set before packing. All packers respect `priority` and `restriction`. `Rotate` is only respected by Monte Carlo packer.

[Back to top](#)

Read-only parameters

```
- state:
"packed" - the part is packed normally
"leftover" - the part could not be packed because the number of the parts was too large for the given tray
"not_packable" - the part is too large for the given tray
"colliding" - the part in its initial position was colliding with a tray wall or with another part, while it was "locked" or the packer option "start_from_current_positions" was active.
"excluded" - the part was excluded from the packing process by the user;
"ignored" - the part was ignored by the packer (for instance, because its mesh was empty or invisible)
"indefinite" - the packing process has not been started yet or was aborted.
```

Example: Reset packing options for all parts

This script accelerates resetting all packing options for all parts present on the platform currently selected in the project tree, something that could take many clicks in the respective GUI dialog.

```
local meshesAll = {}
local root = tray.root
insertMeshesIntoTable(root, meshesAll)

for i, traymesh in pairs(meshesAll) do
    traymesh:setpackingoption('restriction', 'norestriction')
end
```

[Back to top](#)

Parent page: Fabbproject (?guid=NETF-LUA-FABBPJECTOBJ)