

How to change Material Names

version 2.1

Purpose

This manual explains how to change the material names of objects in Sweet Home 3D.

Requirements

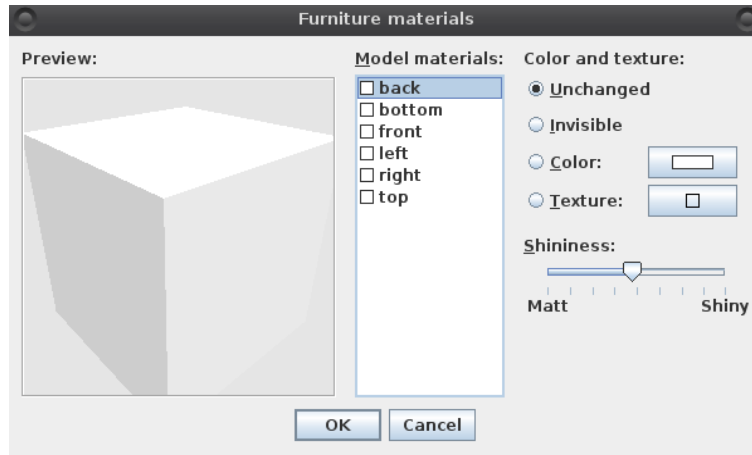
This manual assumes a basic knowledge of exporting and importing objects in Sweet Home 3D and the use of a filemanager in your Operating System. Besides Sweet Home 3D you will need a plain text editor.

Use a new (empty) Sweet Home 3D file to build the examples.

Introduction

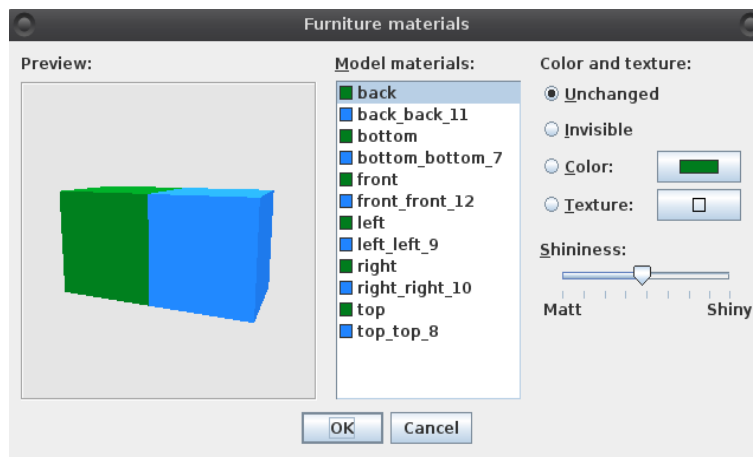
When you drag an object from the furniture list into the 2D window and select *Modify Furniture* and then click the button *Modify* under *Color and texture* you see which material names are available for that piece of furniture.

For example the Box looks like this:



The box has six material names, one for each side: *back*, *bottom*, *front*, *left*, *right*, and *top*.

The box is often used as a part of new furniture. Often multiple boxes and other basic shapes like the Triangle and Cylinder are used to create new furniture. You can combine all kinds of parts but when you export your new furniture the material names of the source objects will remain. It looks like this when you export 2 boxes with different colors and import the combination:



Not only do you have “double” names you also have have lost reference to what is what. Imagine you have created a beautiful new piece of furniture consisting of 10 parts. Without changing material names you will have a long list of meaningless material names.

So, we're going to fix that!

A Tower

We are going to create a tower.

We have a box with grass on top, a dark blue cylinder, and a light blue cone. (The cone object is in the accompanying zip file.)

We want the bottom box to have two material names: *Grass* and *Base*. For the tower itself we want one single material name: *Tower*.

We have two problems to solve: first the bottom box has 6 material names, one for each side, and we want to change that so we have *Grass* for the top and *Base* for the other five material names. Second we have 2 objects (cylinder and cone) that we want to have the same material name: *Tower*.



The box

To change a material name we need to have an *.obj* and *.mtl* file where we can change these names. Drag a box from the furniture list to the 2D window and change its size to width 100, depth 100, and height 10. Click the button *Modify* under *Colors and textures*.

You will see the material names as before: *back*, *bottom*, *front*, *left*, *right*, and *top*. Click one of the names and the side corresponding to that name will blink in the preview. For the box we already know that *top* is the top of the box but for other furniture that will not always be the case so take notice of the material name *top*. Change the texture for the top material to one of the grass textures available in the texture drop down list. In the example I selected *grass darker*. This can be done in a later step if you have written down the material name *top* which you want to change to *Grass* but you will see that setting a texture before export will give you an extra indication of what material name must be changed. With textures you can also change the orientation of the texture. That orientation is preserved when you export an object. With wood textures you might want to rotate one or more sides. You can give the other materials a different color but that's not necessary yet.

Now we export the box. Name it *BoxFloor.obj*. The name is not that important since it's only temporary until we have the complete tower but it's easier to recognize what part the object file refers to.

The *.obj* object file

The object file describes the mesh, the shape of the model. The object file does NOT contain the color or texture of each side, only a reference to a material in the *.mtl* file.

We use a plain text editor to open the resulting *.obj* file:

```
BoxFloor.obj
#
# Generated by Sweet Home 3D
# http://www.sweethome3d.com/
#
mtllib BoxFloor.mtl
g bottom_1
usemtl bottom
v -50.0 0.0 -50.0
v 50.0 0.0 -50.0
v 50.0 0.0 50.0
v -50.0 0.0 50.0
vt 0.0 0.0 0
vt 1.0 0.0 0
vt 1.0 1.0 0
vt 0.0 1.0 0
vn 0.0 -1.0 0.0
f 1/1/1 2/2/1 3/3/1
f 1/1/1 3/3/1 4/4/1
g top_2
usemtl top
v -50.0 10.0 -50.0
v -50.0 10.0 50.0
v 50.0 10.0 50.0
v 50.0 10.0 -50.0
vn 0.0 1.0 0.0
f 5/4/2 6/1/2 7/2/2
f 5/4/2 7/2/2 8/3/2
g left_3
usemtl left
vn -1.0 0.0 0.0
f 1/1/3 4/2/3 6/3/3
f 1/1/3 6/3/3 5/4/3
g right_4
usemtl right
vn 1.0 0.0 0.0
f 2/2/4 8/3/4 7/4/4
f 2/2/4 7/4/4 3/1/4
g back_5
usemtl back
vn 0.0 0.0 -1.0
f 1/2/5 5/3/5 8/4/5
f 1/2/5 8/4/5 2/1/5
g front_6
usemtl front
vn 0.0 0.0 1.0
f 3/2/6 7/3/6 6/4/6
f 3/2/6 6/4/6 4/1/6
```

“Reading” the *BoxFloor.obj* file you will notice the material names as you saw them in Sweet Home 3D.

Notice there are six lines that start with “g “. These are groups defined for the box. In this case there is one group for each side but with other furniture that might be completely different. If you export a *room* you will only have a single group although the exported object has 6 sides.

After each line starting with “g “ you see a line starting with “*usemtl* “. That line is a reference to the material name in the mtl material file. *usemtl top* refers to a block starting with *newmtl top* in the mtl file. We will get back to that.

Remember this: *usemtl* in the .obj file is a reference pointer to *newmtl* in the .mtl file.

We want to change the material name *top* to *Grass* so find the line with *usemtl top* and change it to *usemtl Grass*. We want to have all other names changed to *Base* so change all other *usemtl* lines to *usemtl Base*. Instead of six material names we now have two material names in the .obj file: *Grass* and *Base*.

Save the *BoxFloor.obj* file.

Since the lines with *usemtl* are a reference to the “real” material in the .mtl file we will have to change those names in the .mtl file too.

The .mtl Material file

The *BoxFloor.mtl* file is the file where all materials are defined. Use the plain text editor to open the file:

```
BoxFloor.mtl
#
# Generated by Sweet Home 3D
# http://www.sweethome3d.com/
#
```

```
newmtl bottom
illum 2
Ka 0.4 0.4 0.4
Kd 1.0 1.0 1.0
Ks 0.0 0.0 0.0
Ns 1.0
```

```
newmtl top
illum 1
Ka 0.2 0.2 0.2
Kd 1.0 1.0 1.0
Ks 0.0 0.0 0.0
Ns 1.0
map_Kd BoxFloor_top.jpeg
```

```
newmtl left
illum 2
Ka 0.4 0.4 0.4
Kd 1.0 1.0 1.0
Ks 0.0 0.0 0.0
Ns 1.0
```

```
newmtl right
illum 2
Ka 0.4 0.4 0.4
Kd 1.0 1.0 1.0
Ks 0.0 0.0 0.0
Ns 1.0
```

```
newmtl back
illum 2
Ka 0.4 0.4 0.4
Kd 1.0 1.0 1.0
Ks 0.0 0.0 0.0
Ns 1.0
```

```
newmtl front
illum 2
Ka 0.4 0.4 0.4
Kd 1.0 1.0 1.0
Ks 0.0 0.0 0.0
Ns 1.0
```

Again you can see the material names as you saw them in Sweet Home 3D. Each material name is followed by a number of lines describing the material. (What each line means is outside the scope of this manual. There is a separate manual that explains the contents of MTL files.)

There are six blocks, each starting with *newmtl* followed by a name. Remember which name you wanted to change to *Grass*? If you didn't remember or haven't written down that you wanted *top* to change to *Grass* there is another indication in this specific example: you only added a texture to one material name. You will notice there is only one block with a line starting with *map_Kd* followed by an image file name. If you look at the first line of that block you will see that it is the block for *top*. This is important to remember because with other furniture you might want to have multiple different textures on an object. Using the file name after *map_Kd* and looking at that image in a image viewer you can determine what the name is you wanted to set the material to.

Change the line *newmtl top* to *newmtl Grass*. Now the reference in the object file to *Grass* (*usemtl Grass*) is linked to the material description in the material file.

All other material names should be consolidated to *Base*. We have five material blocks left but we only need one. So remove all blocks except the first one from the five unchanged blocks. Change the *newmtl* line of the remaining block to *newmtl Base*.

You now have two blocks in the material file: one block for *Base*, and one block for *Grass*. Save the BoxFloor.mtl file.

Import and check

Drag the BoxFloor.obj file into the 2D window of Sweet Home 3D. This will start the import furniture dialog. Complete the import.

Choose *Modify Furniture* and click the *Modify* button under *Colors and textures*.

If you did everything right you will now see the two material names *Grass* and *Base*. Changing the *Base* material color will change all sides except the top which is *Grass*.

The tower part

Usually you will want to export and edit each part separately because that makes recognizing the material names easier and there's less chance for a mistake.

For the cylinder and cone we are going to make it easier because we want the tower part to have a single material for both the cylinder and the cone part.

Drag a cylinder from the furniture list in the 2D window. Resize it to width 50, depth 50, and height 100. Drag the cone object from the example files into the 2D window and resize it to width 50, depth 50, and height 50. Set the elevation to 100.

Select both the cylinder and the cone, right-click and select *Modify Furniture* (or use ctrl+E). Set the both the X and Y coordinates to 0 and click Ok. This will align both the cylinder and the cone exactly on top of each other at the location 0,0.

Make sure you still have both the cylinder and the cone selected and export both to *TowerPart.obj*.

Edit the object file: change every "usemtl " line to "usemtl Tower" since we want every part to have that material name.

Edit the material file: Change the first line of the first block to "newmtl Tower". Remove the other blocks.

Save the files and import TowerPart.obj.

Combining all parts

Move all objects in your Sweet Home 3D project away from the 0,0 location. Select the *BoxFloor* and *TowerPart* objects you imported after editing. Move them by setting the X and Y coordinates to 0. Set the elevation of the *TowerPart* object to 10 (the height of *BoxFloor* is 10, this puts the *TowerPart* on top of it). Export both objects to *Tower.obj* and import it again.

Modifying the *Tower* object now shows your tower where you can set different colors or textures to *Base*, *Grass*, and *Tower*. To make it perfect you set the colors and textures you think are best and export the to *Tower.obj* again. Look at the file dates to see which images are used with the latest Tower.Obj and Tower.mtl file. Select them together and zip them as *Tower.zip*.

You now have your custom furniture in a single file ready to use wherever you want.

Next: *Split material names*

Split material names

The Roof Generator is a great tool to generate roofs for your home, but it has one drawback: all borders have the single material name *border*. This was deliberate because having a separate material for each border could result in a huge amount of material names. In normal circumstances this is not a problem but when you apply a wood texture the alignment of the texture is different for each border resulting in some borders displaying a diagonal texture. Since there is only one material *border* you have no way to fix the alignment of a specific border.

But now you know how to manipulate material names! So we are going to fix this for a roof where we want to use a wood texture for the borders.

If you already have a roof generated with the Roof Generator you can use that roof, otherwise create a roof using the Roof Generator tool. Just use a simple roof for this exercise.

Export the roof to *roof1.obj* and open it in your text editor. We are interested in the *border* material so look for the lines with *usemtl border*. You will see multiple lines because all borders use the same material *border*. To allow us to modify each border separately they need unique names so start with renaming all lines *usemtl border* to one with a unique name. The easiest way is to use *usemtl border1*, *usemtl border2*, etc. When you have all lines changed there is no longer a line with *usemtl border*, only lines with *usemtl border* followed by a number.

Save the .obj file.

Now we have to make sure there is a reference to the new material names we added in the .obj file. These are listed in the .mtl file so open the *roof1.mtl* file with your text editor.

There is only one block that starts with *newmtl border* because that is the original material used for all borders. We will have to create new blocks for all the material names we changed in the .obj file. For this example that is very easy: copy the *newmtl border* block and past it for every *usemtl border<number>* we have in the .obj file. Rename them to have a *usemtl border<number>* for each one in the .obj file. You no longer need the original *usemtl border* block because it's not referenced in the .obj file, you probably used that block for *newmtl border1* otherwise remove it.

Save the .mtl file.

Now import the *roof1.obj* file by dragging it into the 2D pane of your Sweet Home 3D project. You now have a separate material name for each border allowing you to align (rotate) the wood texture for each border individually so it displays correctly.

Good Luck!

Update 2024

The roof generator has been greatly improved but this also means that the way to split of the borders has also changed. It still works as described but you can't see some of the borders because they are hidden. These are borders where two sides are attached, mainly at the top and corners. You will still have to rename each border but after import you can see in the Modify furniture dialog which borders are hidden.

Since they are hidden you can't see them, and if you can't see them you don't need them. Make the hidden borders invisible and export and import your roof again. Sweet Home 3D doesn't export invisible parts what makes this action an easy way to get rid of hidden borders. After import the hidden borders are gone. This has two advantages: the resulting object is simpler (less borders) and the object size is reduced in size. With complex roofs both are very welcome, especially the reduction of the number of borders you have to process.

By-the-way, it's not an error that the roof generator creates those hidden borders. If you want to make a side of the roof invisible after generation you will need that hidden border. In that case first make the side invisible, export, import, and you can see which border that was previously hidden is now visible and thus should remain visible. This also means you will have to split the inner materials because making a side invisible will still show the inner part. To allow making just that inner part invisible you will also have to split all inner groups to individual materials.

All in all more work but it gives you a much finer control over the roof materials.