Slave Object Tutorial:
Repository Textures From Maxis Objects

How to make a custom object use the texture and recolors of an original game object

by IgnorantBliss at Mod The Sims 2

Needed for this tutorial:

- SimPE version 0.60 or newer
- CEP (and CEP files also selected in the SimPE File Table)
- A mesh editor and a uv-mapping program if you're going to edit a mesh and/or make a new uv-map for one
- Previous experience on creating new objects and using SimPE in general (the usual steps of creating a new object are not explained in detail in this tutorial)

1. Introduction

This tutorial explains how to make your cloned object borrow its texture and recolors from an original (Maxis) object.

Reasons you might want make your object borrow its textures from an original object:

- You want your new mesh to have matching textures with the in-game object (such as having a curtain borrow its textures from bedding)
- You have only edited the original mesh slightly (made a sofa to match an existing loveseat) or not at all (made a hacked object with new interactions but retained the original object's look)
- You want to keep the file size as small as possible by not having textures included in the package

After you make the object or at least one subset in it borrow its textures from outside of the package, you can not recolor that subset/object the usual way. Instead, you will need to recolor the original game object for the recolors to show up.

You can have a maximum of two recolorable subsets in an object, and this also applies to objects that borrow their textures from elsewhere. You can, for example, have one subset in the object borrow its textures from another object, and a second subset which has its texture in the cloned package as usual. The subset that has its texture, material definition and material override included in the package can be recolored the normal way, while the other subset needs to be recolored by making a recolor of the original game object.

The tutorial assumes you are already familiar with the basics of creating a custom object, and the details of that are not explained here.

The tools needed for this tutorial are the same ones you need for creating a custom object in general, specifically SimPE, a mesh editor such as Milkshape, and possibly a uv-mapping program such as UV-Mapper Classic. A paint program may be needed for viewing a texture for reference, but you won't be doing any texture editing within the tutorial since you will be using the original game textures. Having the CEP installed ensures (with the majority of objects) that the clone is automatically color enabled, and therefore the full process of color-enabling the object is not explained in this tutorial.

As an example in this tutorial I'm going to show how to make a curtain mesh borrow its textures from bedding. You can pick different objects according to your preference, but it might be easier to first get familiar with this method by copying the tutorial exactly.

Important: The object that contains the materials and textures is called the "master" object, and the one that borrows its textures from the master object is called a "slave" object. In my example with the curtain and the bedding, the bed containing the bedding is the master, and the curtain is the slave.

Let's get started.

2. Selecting an object to clone

For the slave object, I suggest cloning an object that is the closest to your desired finished object when it comes to dimensions and function. If you want to make a (3-tile) sofa that borrows its textures from a (2-tile) loveseat, you should clone an existing (3-tile) sofa as your base, slave object. In some cases it's also possible to add new tiles to an object if necessary, but this will not be covered here.

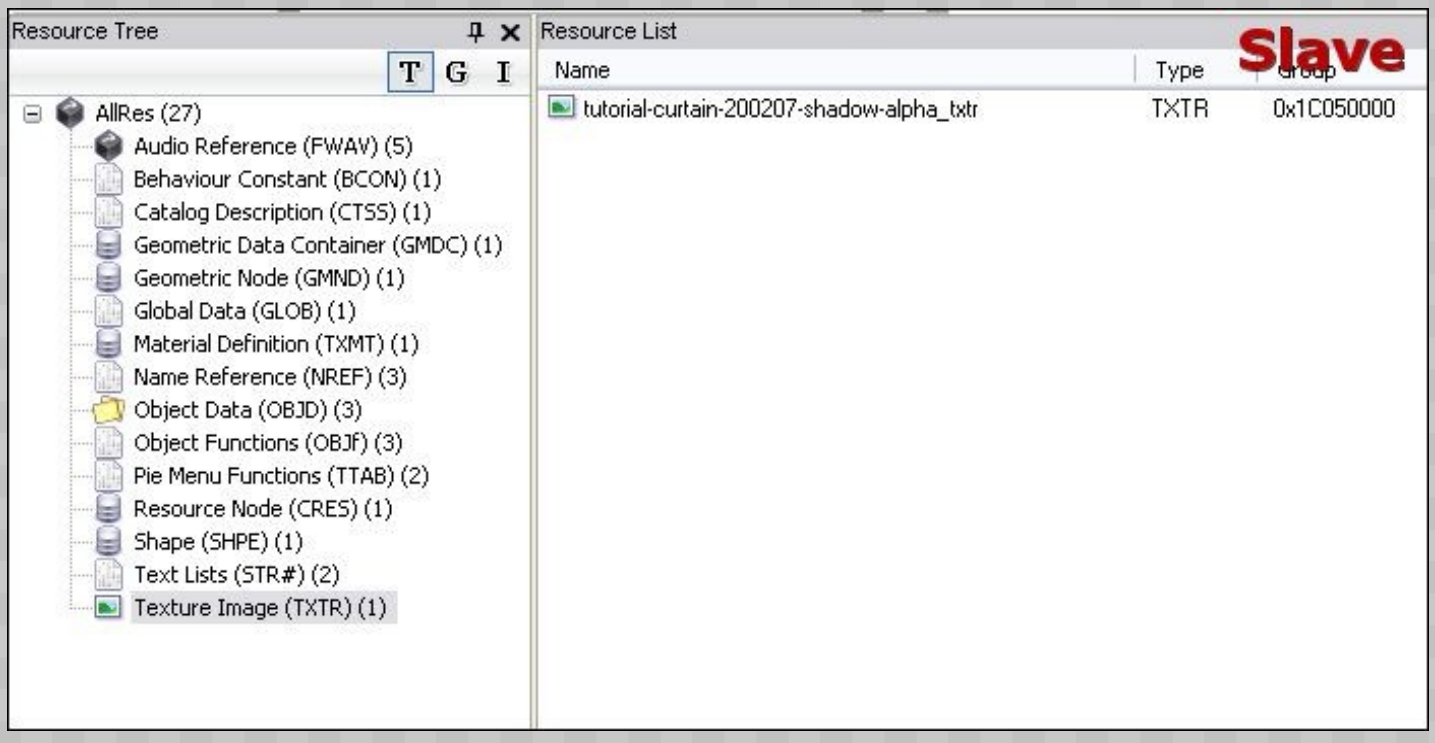
For my example object I'm cloning the 2-tile curtain called "Loft Curtains by Sparse and Fine" from the base game.

3. Cloning and cleaning up the package

a) Clone your object with your usual (preferably the default) cloning options. Give it new GUID(s) as usual.

b) Delete all unnecessary Material Overrides (MMATs), Material Definitions (TXMTs) and Texture Images (TXTRs). You delete files by right-clicking on them in the Resource List and selecting Delete. If there is going to be no subset in the clone that has its textures included in the package, you can delete all of the aforementioned files. If you have one subset that's going to be a slave to another object, and another that has its texture included, then you have to make sure to delete the correct ones. The name of the subset can be seen in the file names of the MMATs and TXMTs that are related to it. The texture file name usually doesn't mention the subset it belongs to, but it should be obvious from looking at the texture image. Often more than one subset uses the same texture.

For my example object, the curtain, I'm deleting the Texture Image, Material Definition and Material Override for the actual curtain texture (subset called "fabric"), but leaving the Texture Image and Material Definition for the shadow into the package since the shadow won't be borrowed from the bed.



4. Making the mesh and the uv-map

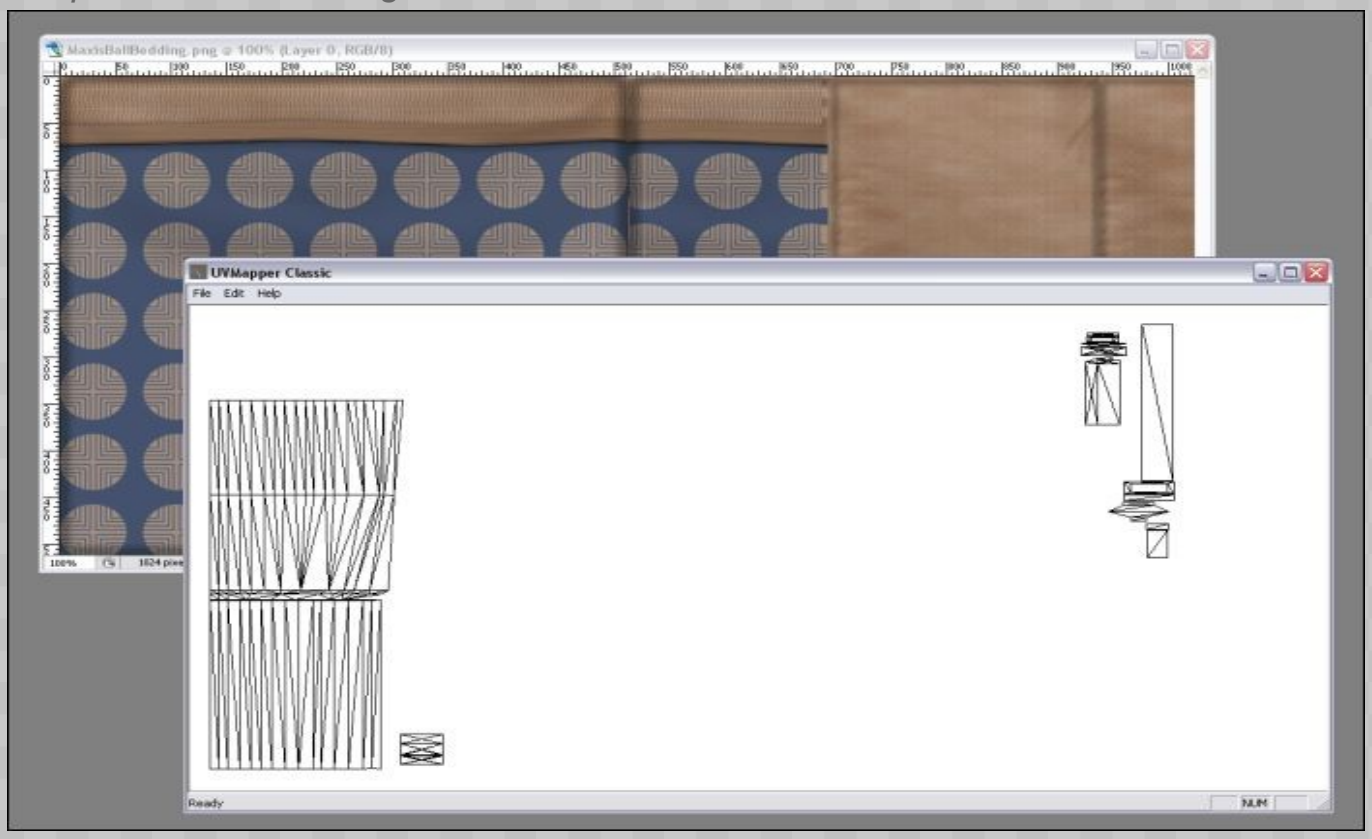
a) Make the mesh as usual in your mesh editor. If you want to use the mesh from the master object as a base for your new mesh, also make a separate clone package at this point of the master object (this clone you can delete once you're done with this tutorial) and export the mesh from that one for editing.

In my example of the curtain, I'm not changing the mesh at all since I want to keep the original shape of the curtain mesh and only change its uv-mapping in the next step.

b) The need for uv-mapping depends on what kind of a mesh you're making. If you're using the master object's mesh as a base and are only making slight (or no) changes to it, there should be no need to adjust the uv-map at all. If you're modying the mesh considerably, or making a completely new mesh, then you probably need to uv-map it, too. For making the uv-map, it is often helpful to extract the texture from the master object to use as a reference for your mapping.

Since you're borrowing the texture instead of making a new one, you will have to adjust your mesh to the texture and not vice versa. For example, if you're making a curtain borrow its textures from bedding, map the curtain mesh to an area of the bedding texture that you want to use, and note that different bedding textures have various patterns, and mapping that looks good with one texture might not look so with another.

This is what the uv-map looks like for my example curtain after I mapped it to use the parts of the bedding texture I wanted.



c) Import your finished mesh to SimPE as usual.

5. Cloning the master object for reference

a) Since we will be linking the new object to an existing one, we need to take a look at the original object's structure and filenames to be able to do that. Normally, when you clone an object with the default options, the file names in the package are different from the original in order to make sure the clone will not override the original when put into the game. However, now we need to make a clone with the original filenames intact. This clone is only created for reference and does not need to be saved.

b) At this point of the tutorial it's helpful to have two SimPE windows open at the same time, one with your cloned slave object, and one with the (temporarily) cloned master object.

c) For the cloning options, unselect every option but the “Pull only default color” one.

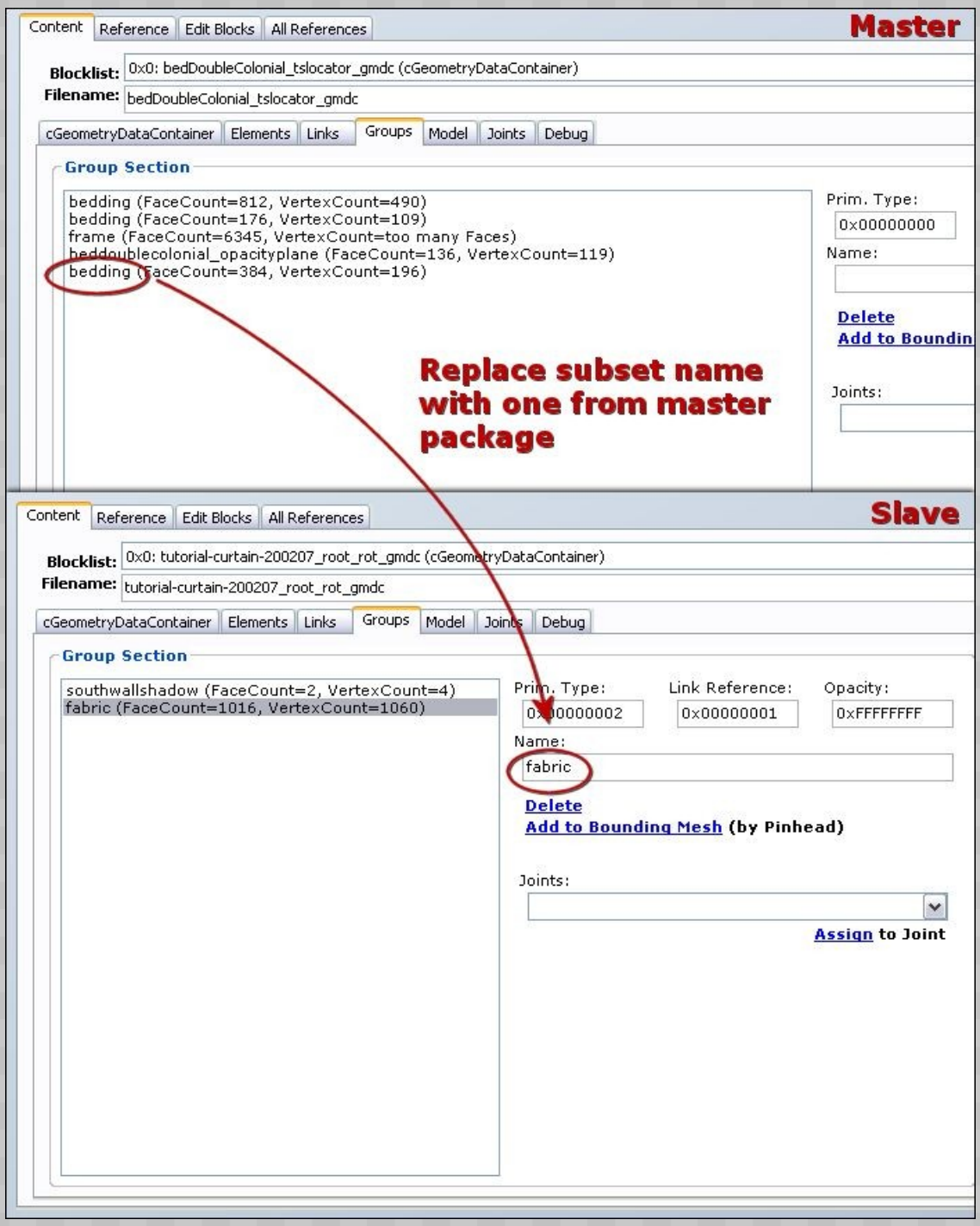
For my example curtain, I'm cloning the Colonial Ironwood Bed since this is the bed that all the other beds in game borrow their bedding textures from.

6. Editing the Geometric Data Container

a) After you have cloned the master object for reference, open the Geometric Data Container (if there are several of them, open the one(s) that contains the subset you want to borrow textures from), and go to the Groups tab under the Content tab. Take a look at the subset names listed.

b) Then open the same tab in your slave object package, copy the subset name(s) from the master package to replace the wanted subset name(s) in the slave package, and Commit and save the slave package. (To edit a subset name, select it from the list by clicking on it and its name will appear in the Name box on the right, where you can edit it.)

For my example object, I'm replacing the curtain's "fabric" subset name with the subset name "bedding" from the master object.

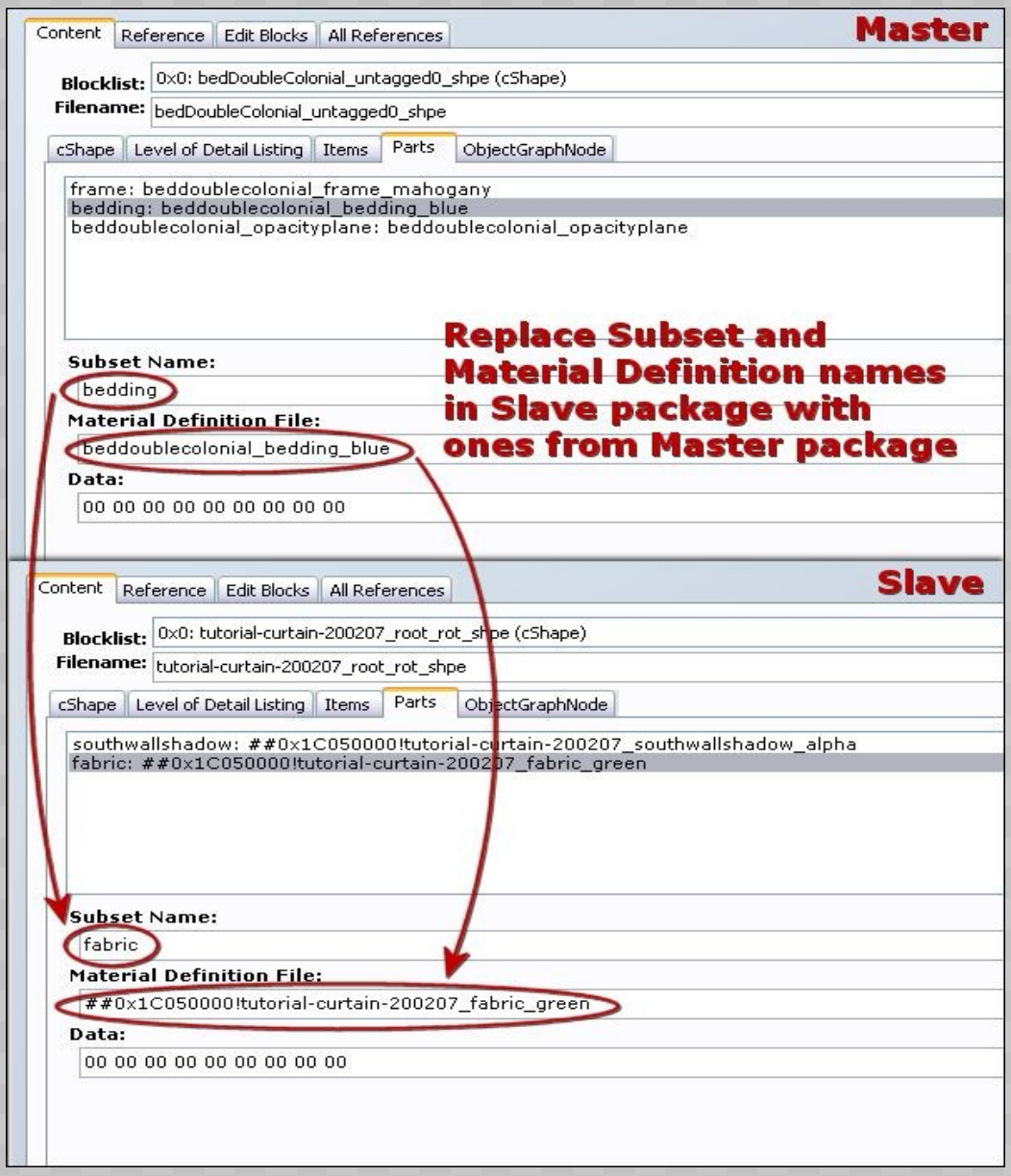


7. Editing the Shape

a) Next, go to Shape and open the Parts tab in both packages. (If there are several Shapes, find the one that contains the subset you're linking. Sometimes you need to edit several Shapes, as is usually the case with objects like windows and doors.)

b) In the master object package, click on the subset you're borrowing the texture from, and copy the text from the box under "Subset name" to replace the Subset Name in the slave package. Then copy the name from the box under "Material Definition File" to the same box of the newly renamed subset in the slave package. Make sure to replace the name completely, don't leave any parts of the old filename there. Commit. If you have two subsets borrowing their textures from elsewhere, repeat these same steps for both of them. Then save the slave package.

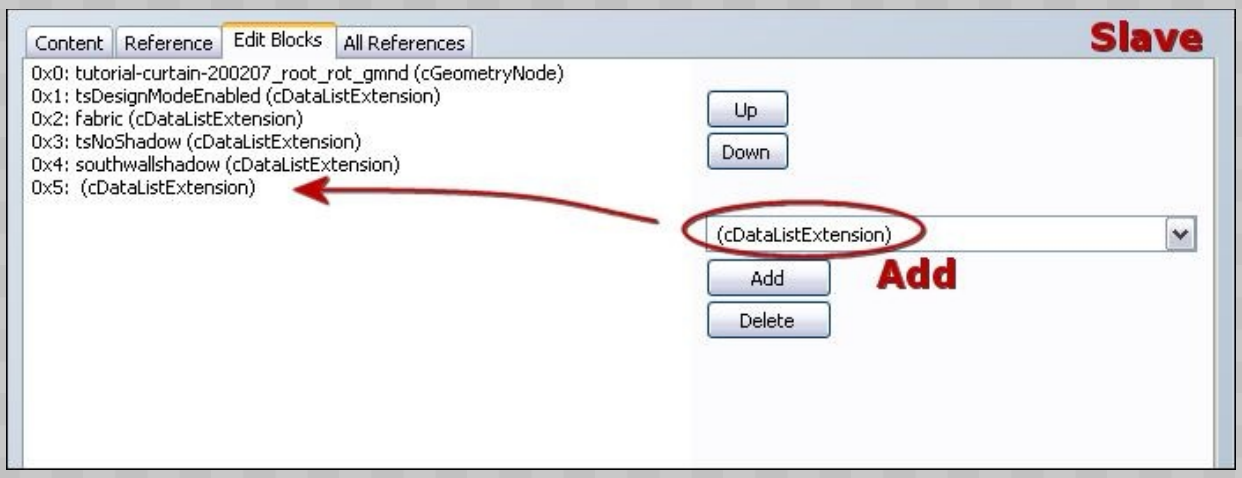
For my example object, I'm replacing the "fabric" subset name with "bedding", and its Material Definition with the bedding's one.



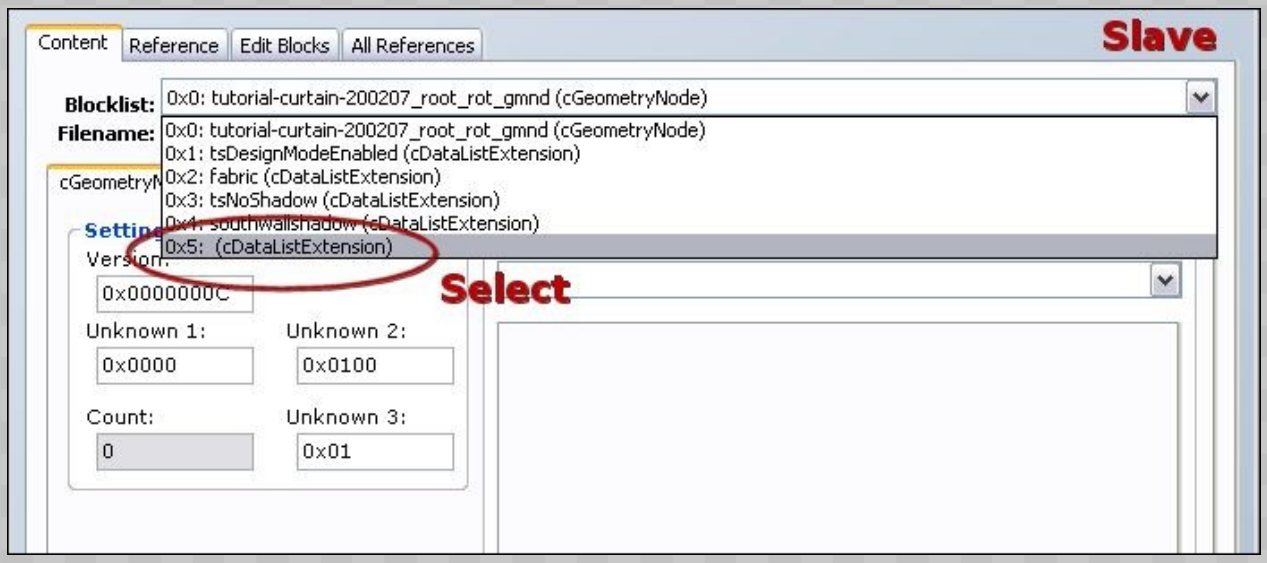
8. Editing the Geometric Node

a) Next, open the Geometric Node in the slave package. Again, there might be more than one Geometric Node to edit, just like with the GMDCs and Shapes. If that's the case, repeat the following for each of them if they contain the subset you're editing.

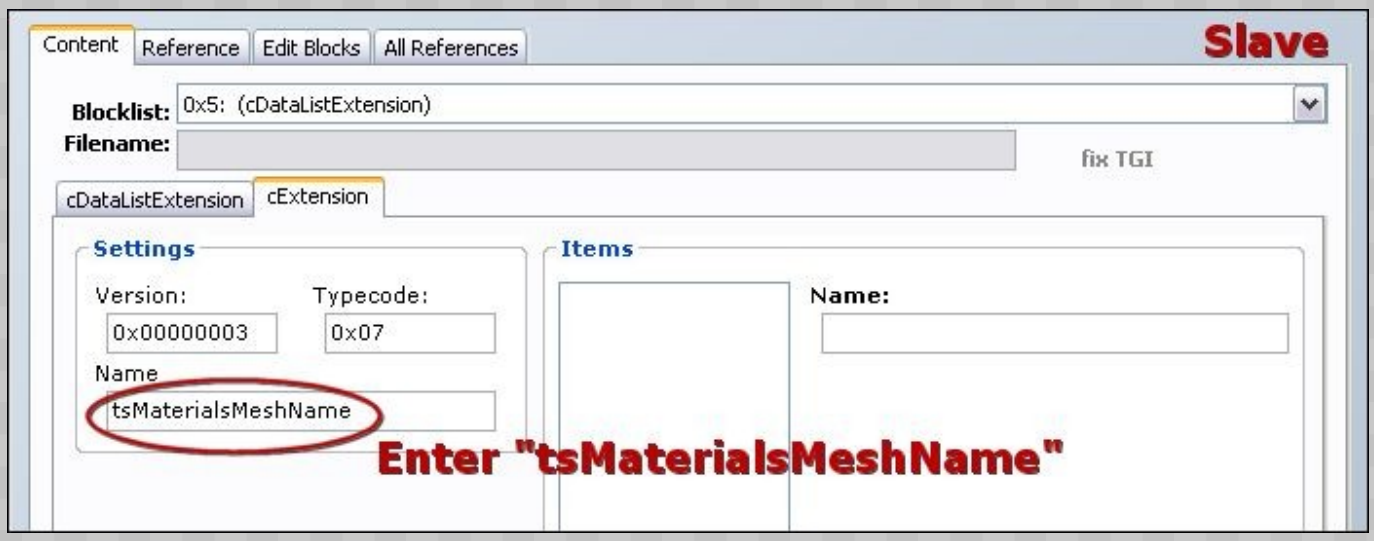
b) In the Content tab under Geometric Node, there is a drop-down menu called Blocklist. Open that and see if there is an entry called “tsMaterialsMeshName”. If there isn't, you need to add one. (If it's there already, you can jump to step c). To add it, click on the Edit Blocks tab, select “tsDataListExtension” from the drop-down menu and click Add.



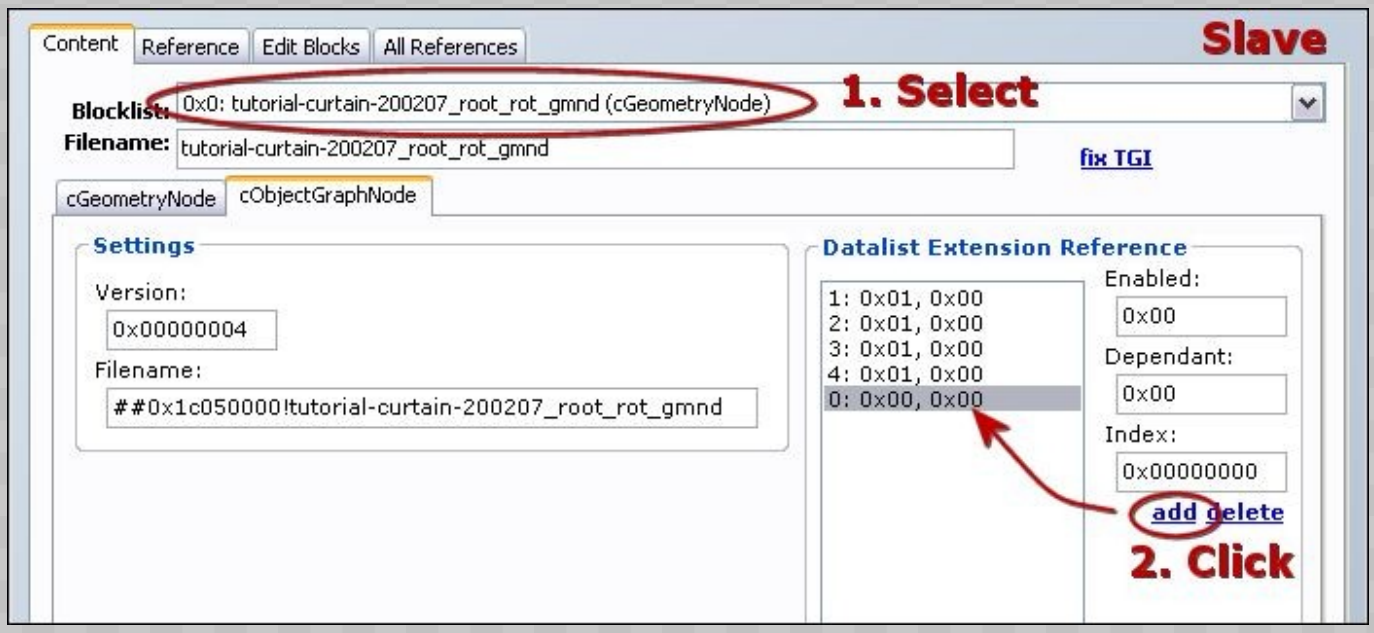
Then go back to the Content tab and open the Blocklist again. Your newly added entry will be at the bottom of the list. Select it.



In the cExtension tab, type “tsMaterialMeshName” (without the quotes) into the Name field and click Commit.

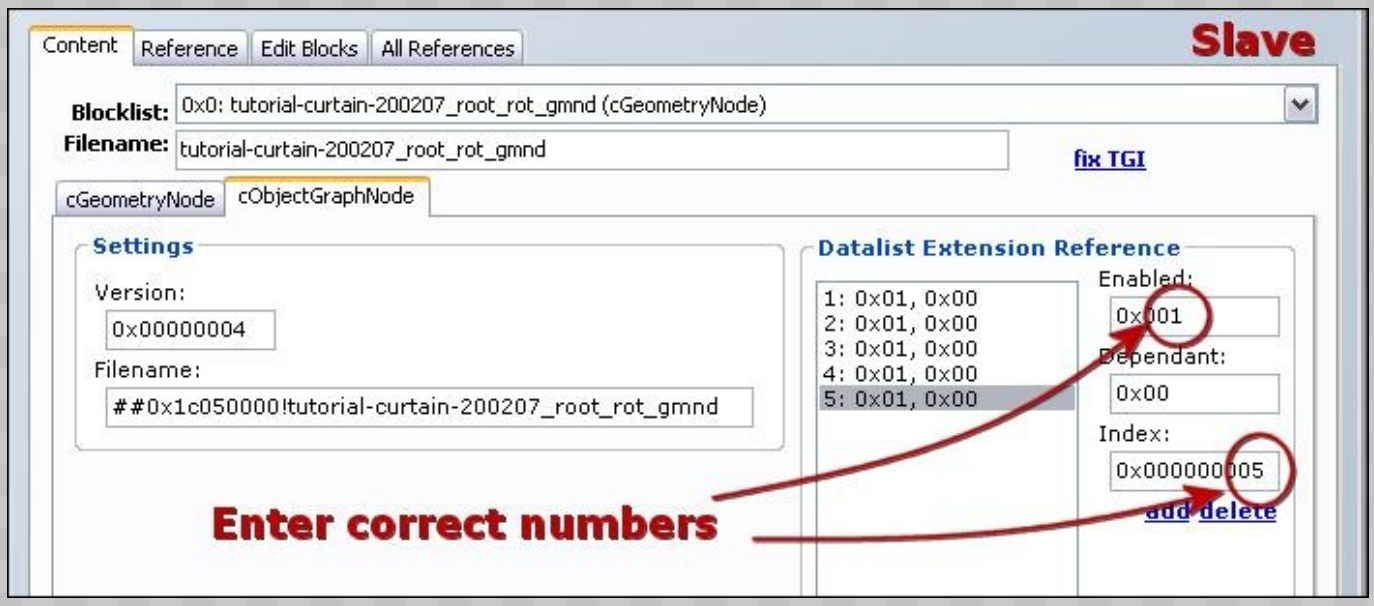


Then select the very first entry on the Blocklist, and go to the cObjectGraphNode tab. In the section under “Datalist Extension Reference” click the “add” button. A new line will appear in the box.

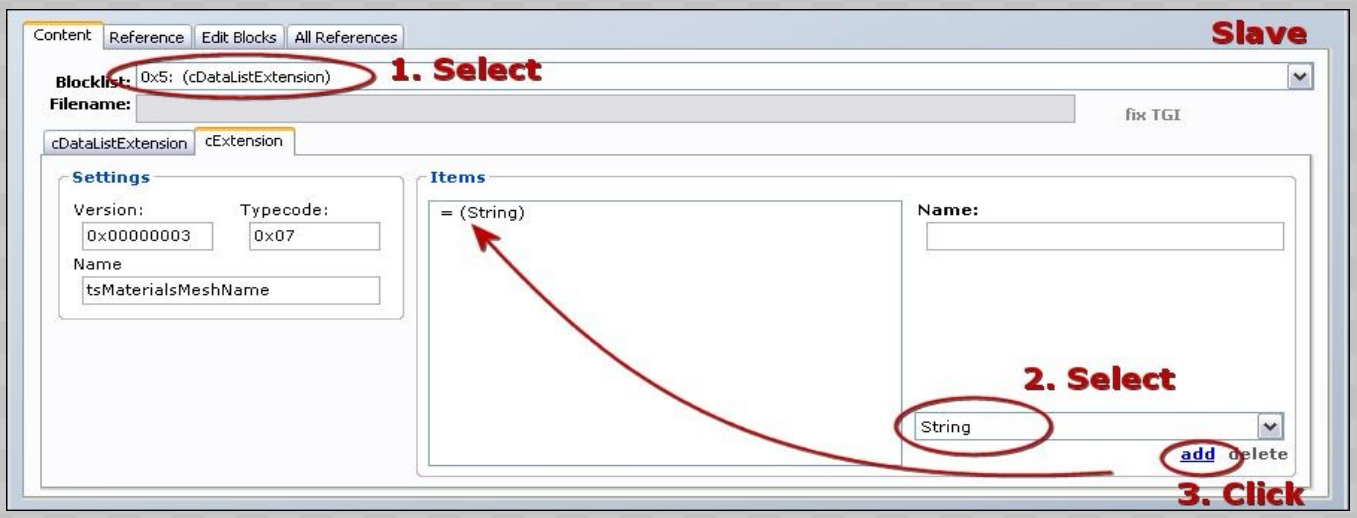


Select the new line, then see which number should come next in row, and add that number into the end of the zeros in the Index box. Then type “1” into the Enabled box, and Commit.

Since for my example object the new entry is 5th on the list, I enter “5” into the Index box.

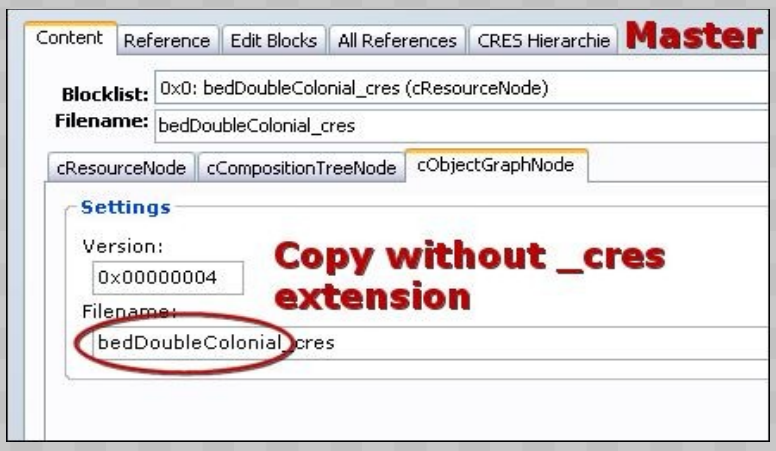


c) After you're done with that (*or* if the tsMaterialsMeshName block already existed) you can move on. Select the tsMaterialsMeshName entry from the blocklist again (the correct name may not show on the Blocklist until you save, exit and re-open the package, but don't worry). From the drop-down list in the bottom right corner, select String and click add. If you have two subsets borrowing their textures, add two Strings and Commit. Leave this tab open.

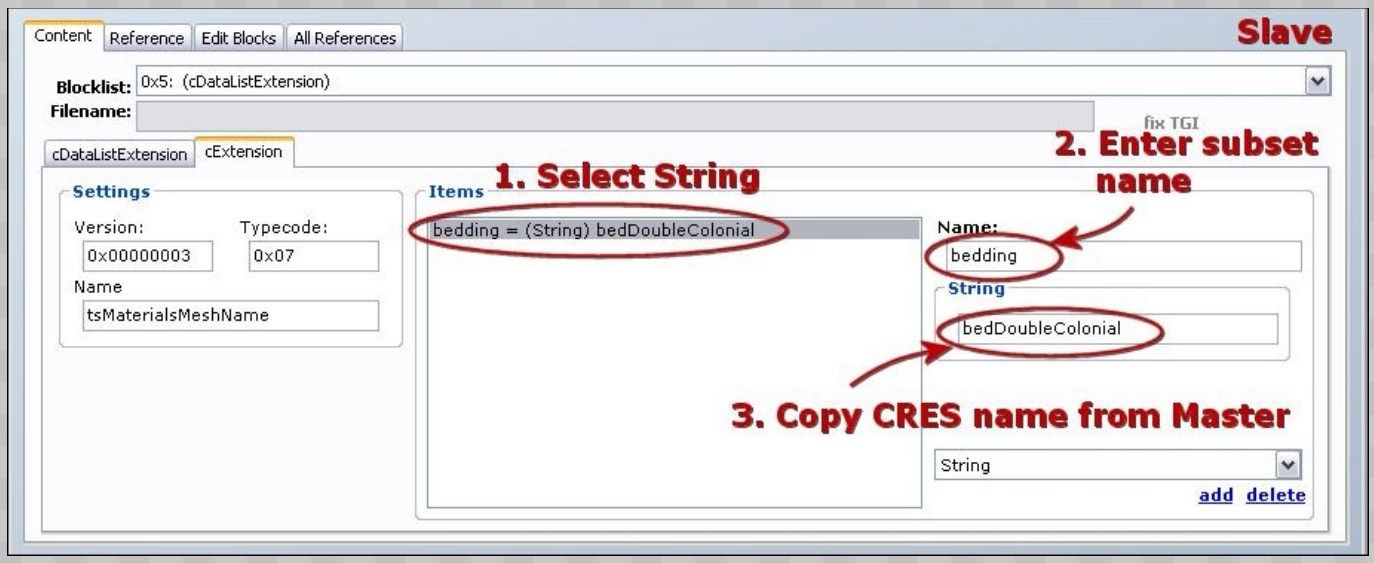


(If your object already had the tsMaterialsMeshName block and it contains one or more Strings already, you may have to delete/replace the original ones with the new ones if you don't want to keep the original links for those listed subsets, especially if subsets listed don't exist in the package anymore after you've edited it. To delete a String, just select it, click Delete in the bottom right corner and Commit.)

d) Go back to the master package, open the Resource Node and select the tab called cObjectGraphNode. Copy the name in the Filename box, *without* the "_cres" extension.

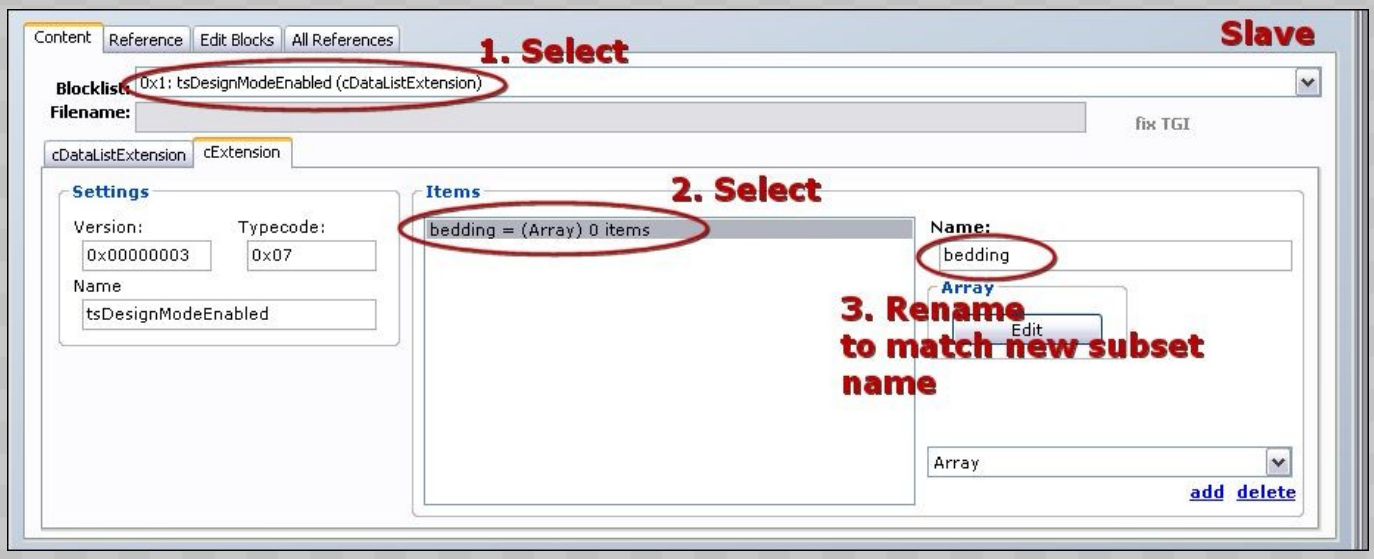


e) Go back to the slave package and the tsMaterialsMeshName block in Geometric Node we have open. Then click on each added String in the Items box, type the name of the subset into the Name field and then paste the copied name from the master object's Resource Node into the String field for each subset. Then Commit and save the package.



f) If earlier in the tutorial you had to change the subset name from the original one (since they need to match those of the master object), there is one more step to do. While still in the Geometric Node, select an entry called "tsDesignModeEnabled" from the Blocklist. In the cExtension tab, in the Items box, you will see the subsets that are listed as recolorable. Rename them to match the new subset names from earlier in the tutorial by clicking on each at a time and editing the name in the Name box on the right, and then Committing. Save the package. (Note that you can have no more than 2 subsets listed under tsDesignModeEnabled)

For my example object, I renamed the "fabric" subset to "bedding".



g) The master package you can close without saving, it won't be needed anymore.

9. Testing your object

Now you can put the slave object package into your game (the Downloads folder where all other custom items go, as well) and see if it works as intended! If not, double-check that you didn't miss any steps in the tutorial. If you still keep having problems, feel free to post questions at the tutorial thread in the forum.