

# How to add capes to your own models:

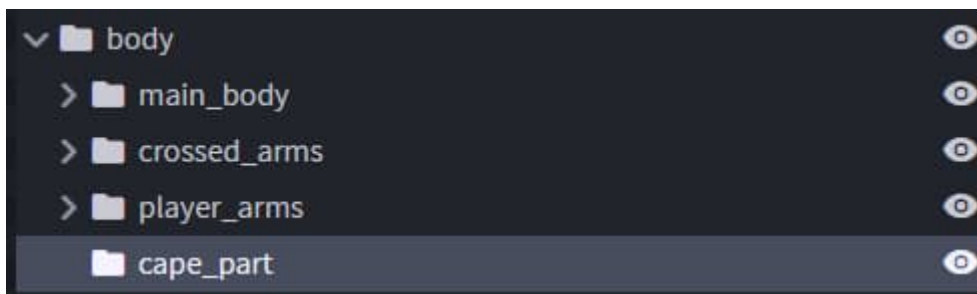
---

## (From 1.0.2 and up)

First and foremost, please use this pack as a dependency, do not reupload it included in your own.

To add the capes to your models, simply create a new bone in the root part of the body (or else if you want funky stuff)

This bone can be named whatever you want.



My blockbench is still not at latest version, so does not support .jpm's. I dk how you set them up, but just make so it read my jpm wich is in `assets\semos\parts` and is called `semos_cape`

Alternatively, open your jem model with a text editor, find the bone you just created and replace it with this:

```
{
  "id": "semos_cape",
  "model": "semos:parts/semos_cape.jpm",
  "translate": [0, 0, 0]
}
```

Reminder "id" can be called what you want

The part should be in a correct location for a player sized model. If not, tweak the location with the translate values

## Get some animations:

You must set animations in a root part. I use this math for a vanilla-ish feel. (limb\_speed might need to be adjusted depending of the mob)

```
"sema_ape.rx": "if(limb_speed>0.6, -  
torad(45),limb_speed>0.1, -torad(15),-torad(5))"
```

You could also reference [\[Semas Animations Lib\]](#) for a smoother anim, more Fresh Animation oriented, the path is:

```
"model": "sema:entity_animations/humanoids/ape_animations.jp"
```

Or make your own :)

## Textures:

The .propertie will make your model use the default weight distribution of all the vanilla capes (+ the classic optifine). It is based on the numbers of real life owner of those capes (according to [NameMC](#) ) It will also have about half a chance to not have any cape

There are a couple of variables you can set in you model to force some king of capes. Those variables must be booleans and named the following

- `extra_rare_capes` (those with less than 10 owners)
- `rare_capes` (those with less than 500 owners)
- `common_capes` (those with more than 100k owners)
- `minecon_capes` (those from Minecon 2011 - 2016, Experience and Minecon Live 2019)
- `no_capes` (Never display a cape)

When true it will follow the weight distribtuion of only the corresponding capes