

Cube - Custom Utility and Blocks Engine

Before Anything

Note: this part is the most important, without this, nothing will work.

You first need to make a loot table. And make sure to name it your exact block. If not, it will break everything

1. Creative Inventory Blocks

Before anything, I'd like to thank KubaKu for letting me use their font.

This part of the engine basically makes all your custom blocks be in the creative inventory, like this:



The custom block in the creative inventory is just a painting that we will replace with the help of the loot table. You will need to make a custom painting in a `painting_variant/blocks` folder.

Make sure the name of the font is the exact name of the loot table.

The font should look like this:

```
{
  "title": {
    "font": "omg_font:custom",
    "translate": "painting.example_pack.mycustom_block"
  },
  "width": 16,
  "height": 16,
  "asset_id": "example_pack:blocks/mycustom_block"
}
```

```
}
```

Now in the minecraft en_us.json folder, we will need to have something like this:

```
"enchantment.omg_font.ex_one":  
"\uE600\uE601\uE602$fExamples\nTest",  
  "enchantment.omg_font.ex_two":  
"\uE600\uE601\uE602$fABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890",  
  "painting.dimensions": "",  
  
  "painting.example_pack.mycustom_block": "\uE60A  \uE600\uE606  
\uE604\uE606  \uE603  \uE606  \uE603  \uE606\uE603  \uE606\uE603  
\uE606\uE601\uE606  \uE605  \uE609\uE608\uE608\uE608\uE608\uE606  $fMy  
Custom Block"
```

the first 3 lines you do not need to touch them, only the last. The font is there to have the background like in this image



Now we need to give the painting a texture, so we go into the minecraft items folder into painting.json

```
{
  "model": {
    "type": "minecraft:select",
    "property": "minecraft:component",
    "component": "minecraft:painting/variant",
    "cases": [
      {
        "when": ["example_pack:blocks/mycustom_block"],
        "model": { "type": "model", "model": "example_pack:block/mycustom_block" }
      },
      {
        "fallback": {
          "type": "minecraft:model",
          "model": "minecraft:item/painting"
        }
      }
    ]
  }
}
```

“when” is the painting variant folder.

“model” is the model we want.

“fallback” is the original painting for all the other paintings we don't need a custom texture for.

2. Custom Blocks, no entities

This part of the engine gives 3 custom block methods:

1. Double Slabs Blocks.

This method is the most recommended because it has no lag. If you don't want an entity with it, this is the best one. Because it's a block state, you can see the breaking animation and the ambient occultation.

2. Waterlogged Double Slabs Blocks

This is similar to the other one, but you can't find it in structures. The other method makes double slabs an unused block state, but this doesn't need anything to make it an unused block state. When breaking the block, water will flash for 1 tick and you will see water dripping from the block. As the other one, it's lagless you want an entity, you can see the breaking animation and the ambient occultation.

3. Item Display Blocks.

If you use this method, you must use as a base block a spawner. This is not lagless if you don't want it to have a use. You can't see the breaking animation and the ambient occultation.

Now that you chose your custom block method, you will need to create a folder in `example_pack:blocks/mycustomblock`. Make sure the folder name is the exact same as the loot table.

Then in the function `example_pack:blocks/mycustomblock/placed.mcfunction` you will put the next thing:

```
# method 1:
setblock ~ ~ ~ myslab[type=double]

# method 2:
setblock ~ ~ ~ myslab[type=double,waterlogged=true]

# method 1/2:
summon marker ~ ~ ~
{Tags:["mycustom_block","redstone-powered","piston-able_blocks"]}

# method 3:
setblock ~ ~ ~ spawner

summon item_display ~ ~ ~
{Tags:["mycustomblock",item_display_block,"redstone-powered"],item:{id:
"stick",components:{item_model:"painting","painting/variant":"smarter_b
locks:blocks/block_breaker"}},transformation:{left_rotation:[0,0,0,1],r
ight_rotation:[0,0,0,1],scale:[1.0001,1.0001,1.0001],translation:[0,0,0
]},brightness:{block:0,sky:15}}
```

If you don't want your block to be redstone powered, remove the tag. But don't remove the `piston-able_blocks` tag, that's crucial. We will need it later.

Now in `example_pack:blocks/blocks.mcfunction` you will need to put the next thing:

```
# mycustom_block
execute as @e[tag=mycustomblock] at @s run function
example_pack:blocks/aa_check_blocks/mycustomblock
```

If you use method 1 or 2, in `example_pack:blocks/mycustomblock/main.mcfuction` you will need something like this:

```
execute unless block ~ ~ ~ myslab[type=double] unless score @s delay
matches 1.. run scoreboard players set @s delay 4
execute as @s[scores={delay=1}] run function
cube:piston-able_blocks_engine/check_around
{block:"blackstone_slab[type=double]",tag:"block_hardener"}

#only if you have the redstone-powered tag
execute as @s[tag=powered,tag=!working] run function
example_pack:blocks/mycustomblock/activated
```

And if you use method 3, you don't need to do anything, the block when broken will give you the block back. Only use main if you want functionality for your block.

3. Redstone-Powered Blocks

This part of the engine makes a block do something when any type of redstone is activated, just like any other redstone blocks.

For it to be a redstone block, you need the tag when summoning it "redstone-powered".

The first and only thing we need to do, is go to
`example_pack:blocks/mycustomblock/main.mcfuction`

and put this is if it isn't already there:

```
execute as @s[tag=powered,tag=!working] run function
example_pack:blocks/mycustomblock/activated
```

In `example_pack:blocks/mycustomblock/activated`, you need to put

```
tag @s add working
```

After you can put whatever.

If you want the block to work only in specific situations, you will need this next thing:

```
execute <your conditions> run return run function
example_pack:blocks/mycustomblock/deactivated
```

And in there you will need to putS

```
tag @s add didnt_work
```

After you can put anything. Once any blocks that give power aren't in a 1x1 radius besides the corners of the block, the tag will get removed and everything will go back to normal.

4. Piston Able Blocks

If you use the first method of custom blocks (double slabs) or using any blocks that the piston can move as base blocks, so that excludes any data-based block or a spawner, you will want this. Because when the block is moved by the piston, without this, the entity will die so you will have to replace the block.

The entity should have the tag:

```
"piston-able_blocks"
```

Now, if you use method 1, in example_pack:blocks/mycustomblock/main.mcfuction you will need something like this:

```
execute unless block ~~~ myslab[type=double] unless score @s delay
matches 1.. run scoreboard players set @s delay 4
execute as @s[scores={delay=1}] run function
cube:piston-able_blocks_engine/check_around
{block:"blackstone_slab[type=double]",tag:"block_hardener"}

#only if you have the redstone-powered tag
execute as @s[tag=powered,tag=!working] run function
example_pack:blocks/mycustomblock/activated
```

Where it says block:"", you have to put your base block, if it's a double slab, you'll have to also add [type=double].

Where it says tag:"", you have to put your entity's base tag.

How it works is that it waits for the block to fully be piston-ed (takes about 3 ticks) and then it checks.

Do note that if you use method 2, you won't be able to push it, because the block will transform in a double slab, and it will not be waterlogged, and the entity inside it will die.

5. Note

Here i haven't explained how everything works, it will take too much time for just one google doc. If you want to read the code for yourself, go in the datapack or resource pack and look at the part you are interested in.

All the datapack/resource pack was written by me (ByApac) besides the font which KubaKu gave me permission to use.

Please if you use **Cube** for your project, do not remove any credits or go around advertising it like it's yours.

Thank you for choosing Cube.