

Memory Programmer

This chapter contains information about the Memory Programmer and also how to add new recipes using Datapacks.

- [Memory Programmer](#)
- [Creating Memory Programmer Recipes and Categories](#)

Memory Programmer

The Memory Programmer is a machine that uses energy to program Flash Memory. For example you can program various Void Miner programs onto them in order to make your Void Miner mine different items.

The Interface and Programming



By default the interface displays Categories

of programs and you'll need to select a category before selecting a program. If you need to go back to the Category selection screen you can press the arrow button in the top left of the GUI.

Before you start programming your Flash Memory in the Memory Programmer you will need to supply it with some energy.

To program flash memory all you do is place a Flash Memory Item in the top right slot of the interface and then select a program you want to program to it. Once you have selected a program the Memory Programmer will program the selected program to the Flash Memory and then transfer it to the bottom right slot inside the Interface. Once your Memory is in the bottom slot it is ready to take out and use. Flash Memory can be written over which means you can use your previously programmed memory to add a new program too.

Creating Memory Programmer Recipes and Categories

In this article we will look into how to make Memory Programmer Categories and Program recipes.

Memory Programmer Categories

Folder Location: `data/NAMESPACE_HERE/envirocore/memory_programmer_category/`

This is a base blank category without item Icons:

```
{}
```

Yup that's right, all you need is an empty json file for the base category although you will need to add Localizations to the category id using some sort of resource loader or resource pack. Below is what you'd need to add to your lang file to add a localization to the category:

```
"memory_programmer_category.NAPESPACE_HERE.NAME_OF_CATEGORY_JSON_HERE": "NAME OF CATEGORY  
HERE"
```

Category Icons

You can add item icons to your Category too, below is an example of a Memory Programmer Category with icons:

```
{  
  "icons": [  
    {"raw:item": "minecraft:diamond_ore"},  
    {"raw:item": "minecraft:gold_ore"}  
  ]  
}
```

Memory Programmer Recipes

Folder Location: `data/NAMESPACE_HERE/envirocore/memory_programmer/`

This is the base Memory Programmer recipe.

```
{
  "categories": [],
  "program": "",
  "r": 255, "g": 255, "b": 255,
  "input": {
  },
  "output": {
  },
  "duration": {"raw:int": 40}
}
```

Categories

Setting which categories this shows up under is pretty simple all you have to do is get your category id which would be `"NAMESPACE_HERE:MEMORY_PROGRAMMER_CATEGORY_JSON_FILE_NAME_HERE"`

So for example if you wanted to add your recipe to both your own category and one from environmental tech you could do the following:

```
"categories": [
  "NAMESPACE_HERE:MEMORY_PROGRAMMER_CATEGORY_JSON_FILE_NAME_HERE",
  "envirotech:ores"
]
```

Program ID

The program id would be for example the id of the void miner program.

```
"program": "envirotech:crystal"
```

Flash Memory RGB

The R, G and B values are for the color that the memory will show up as after programmed. Each value can be between the values of 0 and 255.

Input and Output

The syntax for input and output items can be found in: [Recipe Item Types](#)

You will most likely want to just use the Flash Memory item that is in envirocore so the format for that would be:

```
"raw:item": "envirocore:flash_memory"
```

Overriding Recipes

If you want to disable or replace any of the recipes for the Assembler you can follow [This Tutorial](#)