

## [Features](#)

[Overview](#)

[Skin Pack tab](#)

[Menus and settings](#)

## [Tutorials](#)

[Creating a New Project](#)

[Adding a World to Your Project](#)

[Add a Skin Pack to Your Project](#)

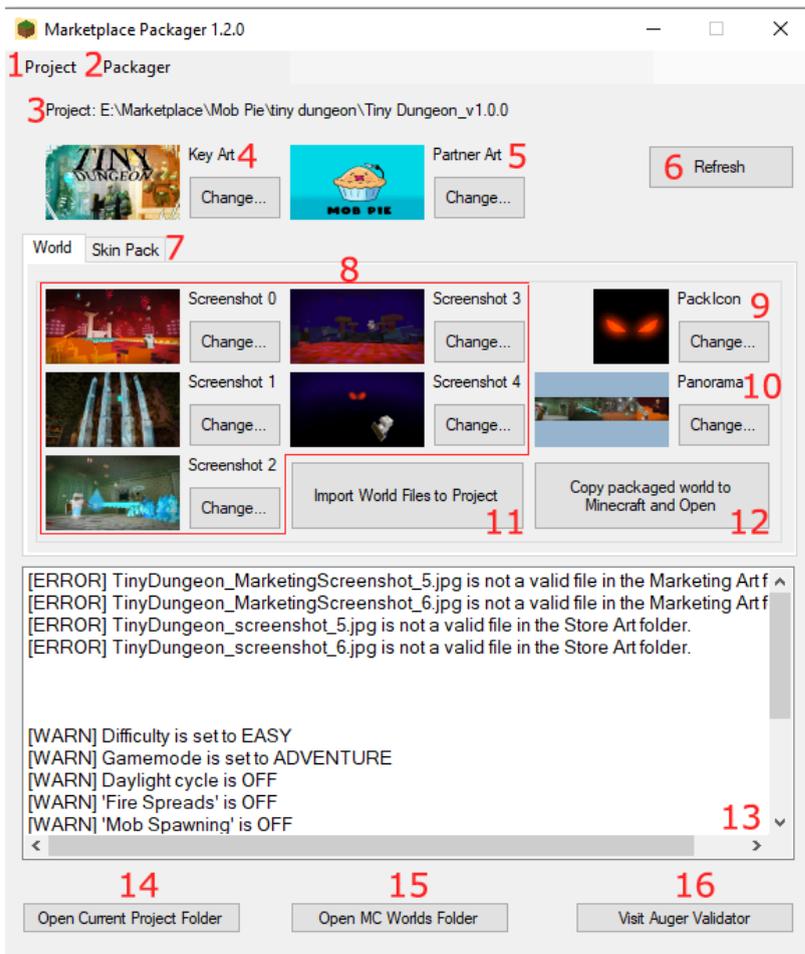
# Map Packager 1.2.0

Documentation version 1.0

This document includes documentation of features, and tutorials for the Map Packager app.

## Features

### Overview



1. Project menu opens projects and creates new ones.

2. Packager menu lets you change the packager settings and check for updates.

3. File path of the project that is currently loaded. Use 14 to open this path in file explorer.

4. Select a key art to add to this project

5. Select a partner art to add to this project

6. Refresh Map Packager if you make external changes to the project.

7. World/Skin pack tabs. See the next image for an overview of the skin pack tab

8. Select 5 screenshots to add to this project

9. Select a pack icon to add to this project.

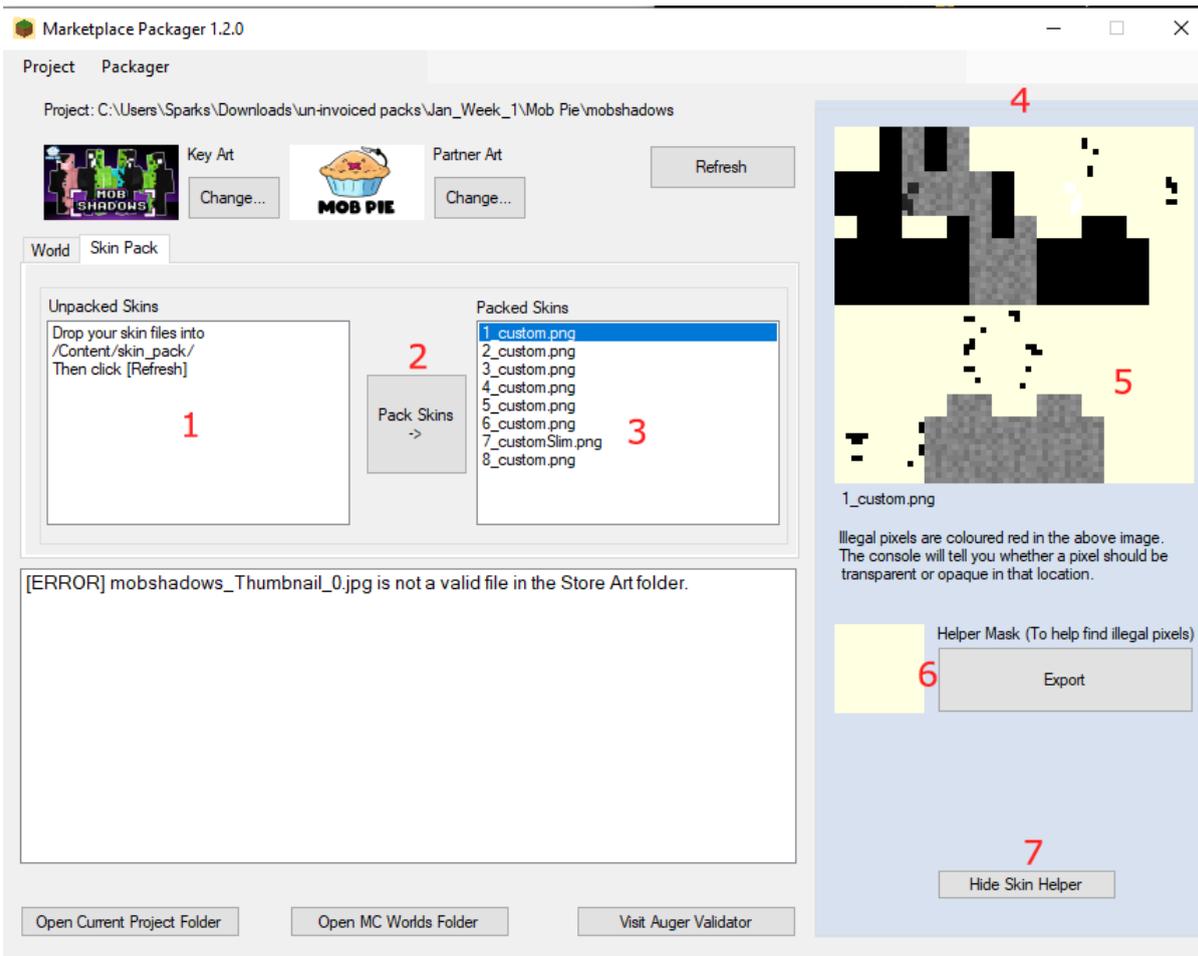
10. Select a panorama to add to this project

11. Browse for a Minecraft world to import the world straight into the project.

12. Create a .mcworld file for this project and launch Minecraft with the world imported.

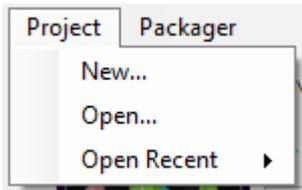
13. Error and warning messages letting you know about the status of your project.
14. Opens the current project files in Windows file explorer.
15. Opens the mcworlds folder where your Minecraft worlds are stored
16. Opens Auger in your default browser. Doesn't give access to Auger if you don't have it, it's just a shortcut.

## Skin Pack tab

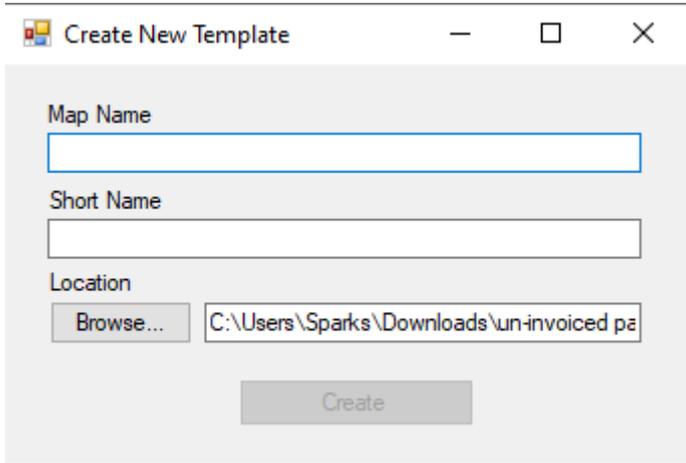


1. A list of skin textures Map Packager finds inside the skins folder that aren't listed in the skins.json file yet
2. Press this button to create the files necessary to add any unpacked skins, moving them from Unpacked Skins to Packed Skins
3. A list of skins that have been successfully added to the project. Clicking any skin in the list will open the Skin Helper (4)
4. The skin helper is a tab that shows you any illegal pixels in your skin texture.
5. A greyscale view of the selected skin, with illegal pixels marked in red to help you spot them. This will not ALWAYS catch illegal pixels that Auger validator notices, but flags the vast majority.
6. The helper mask is a transparent PNG that can be saved, then dropped on top of your skin in your image editor to help locate illegal pixels.
7. Hide the skin helper tab again.

# Menus and settings



- **New:** Opens the “Create New Template” window shown below..
- **Open:** Browse to a folder on your computer containing a packaged project
- **Open Recent:** a list of the 10 most recently opened projects.



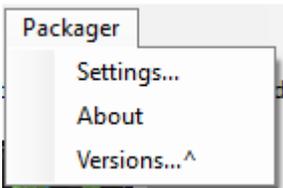
- **Map Name:** The name of your project, exactly as you would like it to appear on the Marketplace

- **Short Name:** This field is automatically generated as you enter a name but can be edited if needed. It is used by the project's internal packaging to keep file paths within required lengths.

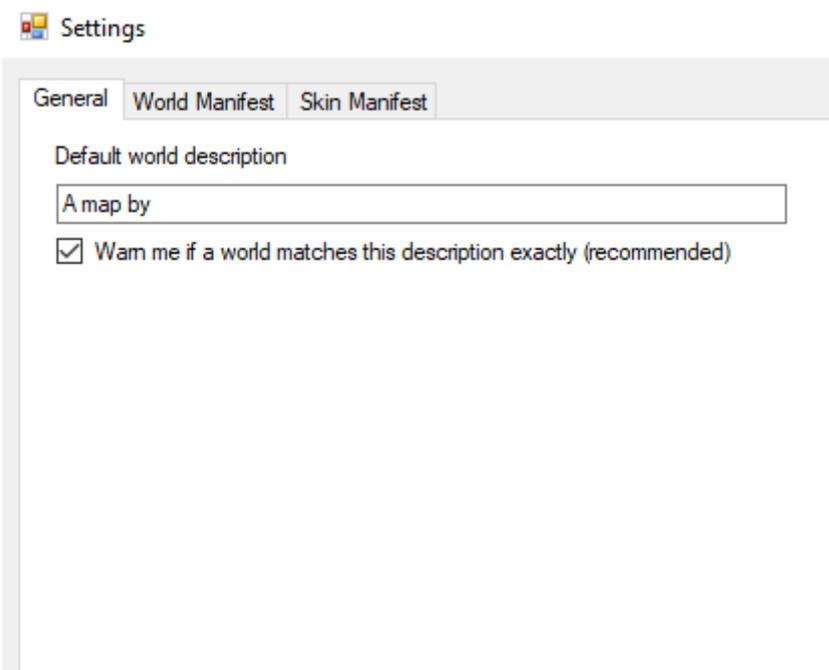
- **Location:** Enter a folder location where you would like the project to be created or click browse to navigate to a folder. The project will

make a new sub-folder within the selected directory to store the project files.

- **Create:** Create the new project. This becomes clickable once you have a name and short name.



- **Settings:** Opens the settings window shown below
- **About:** Information about this program
- **Versions...^:** Opens the Map Packager web page in your default browser.



## General Settings

- **Default world description:** text to automatically fill into the description field of this project. This may be empty by default.

- **Warn me if...** Will display a warning in Map Packager if the project has this exact text as its description (to stop you forgetting to change it)



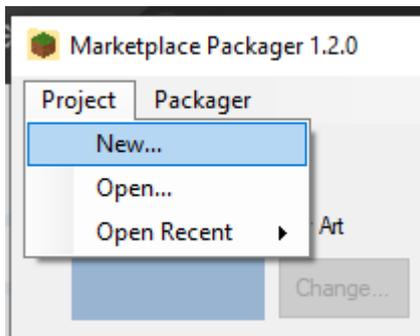
## World/Skin Manifest Settings

- **Template:** A textbox where you can make changes to the manifest template used to create new projects. Generally you don't need to touch this except to update the `base_game_version` to match Minecraft when Minecraft gets an update.

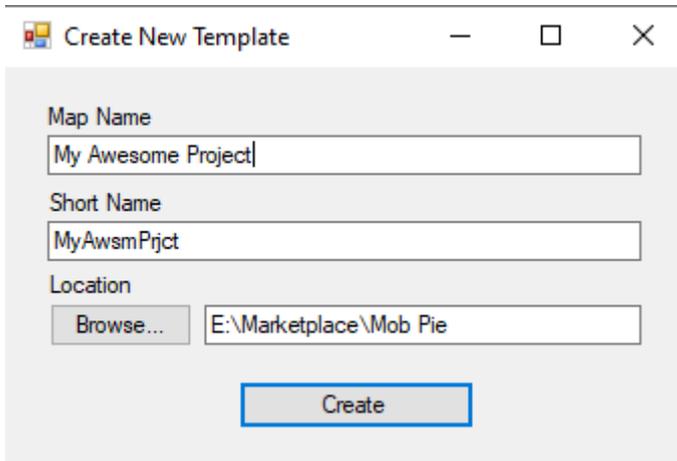
## Tutorials

Projects can contain worlds, skins, or both. Projects are created the same way to begin with, then a world or a skin set can be added to the project using their respective tabs.

## Creating a New Project



Open up Map Packager and in the top right, click **Project** then **New...**

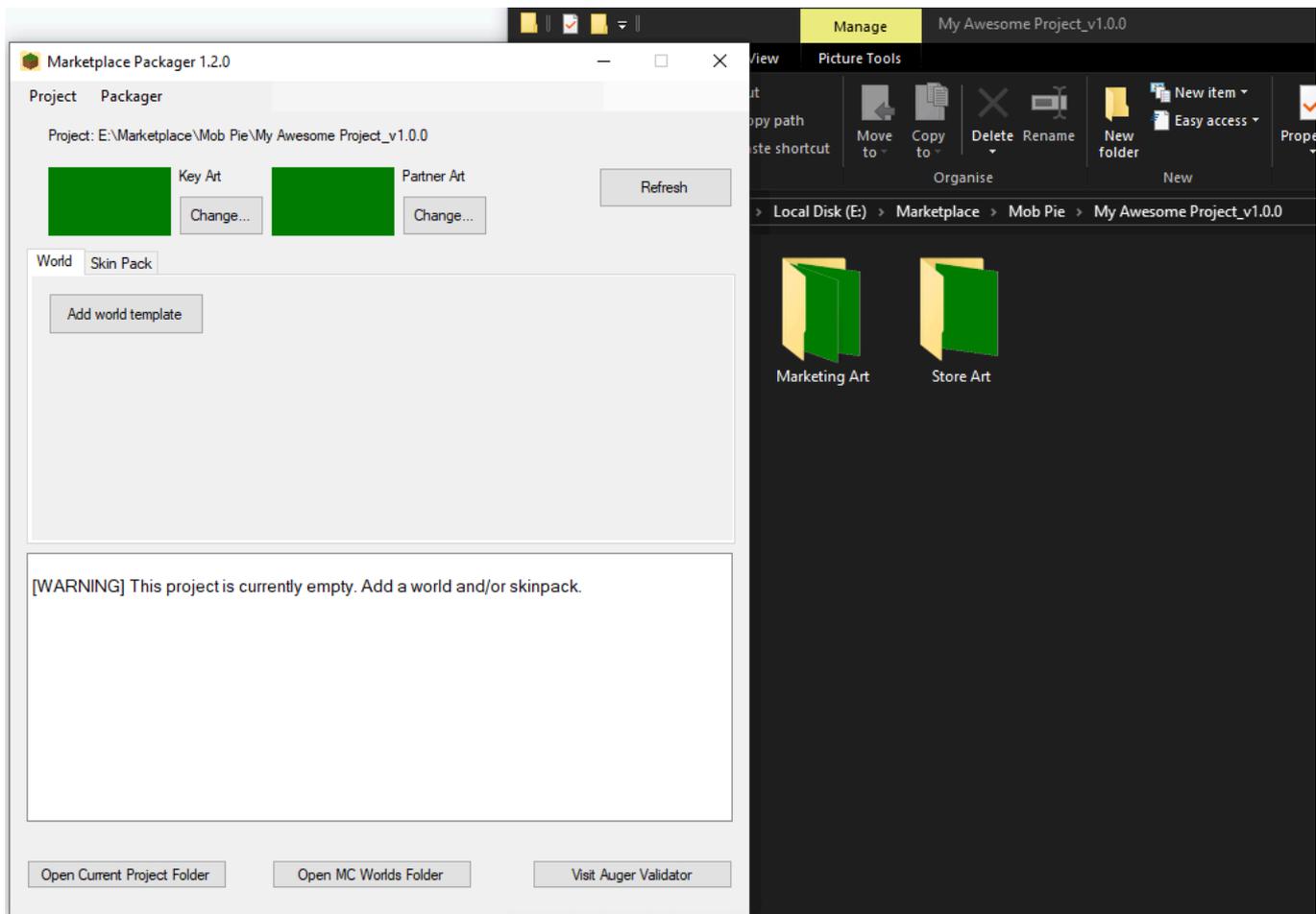


Enter a project name (Map Name) exactly as you wish the offer to appear on the Marketplace.

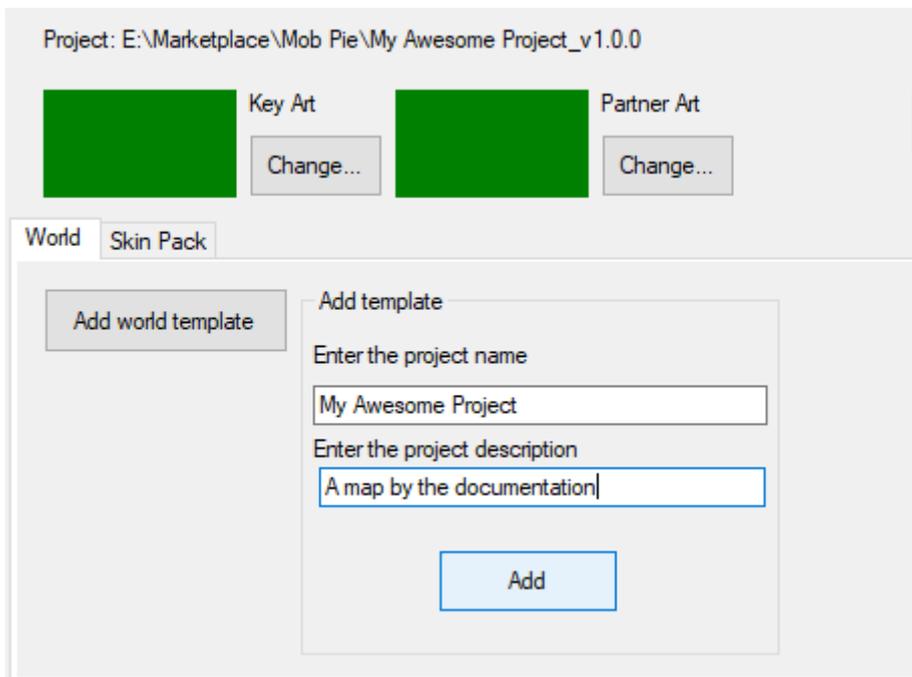
The short name will automatically generate as you type the name. You can edit the suggested short name, but I recommend you leave it as it is. It is only used internally and players will not see it.

Choose a location for your project files to appear. The project will make a new folder inside that location to store everything. Then click **Create**

You'll see that a blank project has been created called My Awesome Project\_v1.0.0. Some files have been generated in your project folder, with blank, placeholder images in the marketing and store art folders. There is a **warning** that the project is empty. You'll need to add a skin pack, world, or both to create your offer.



## Adding a World to Your Project



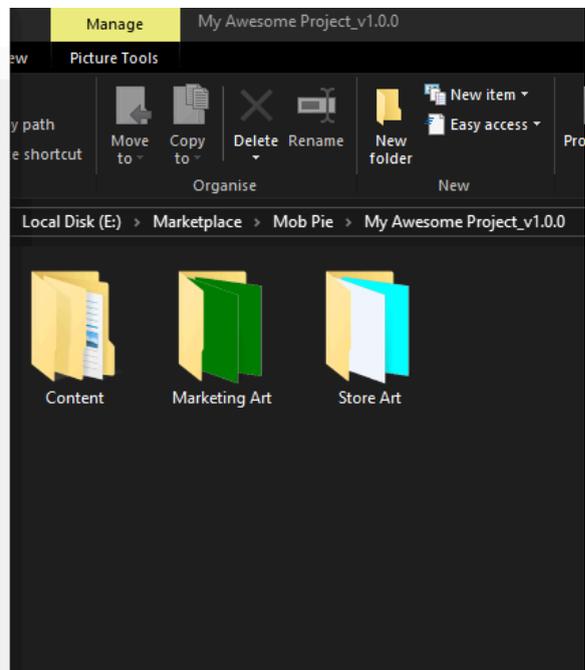
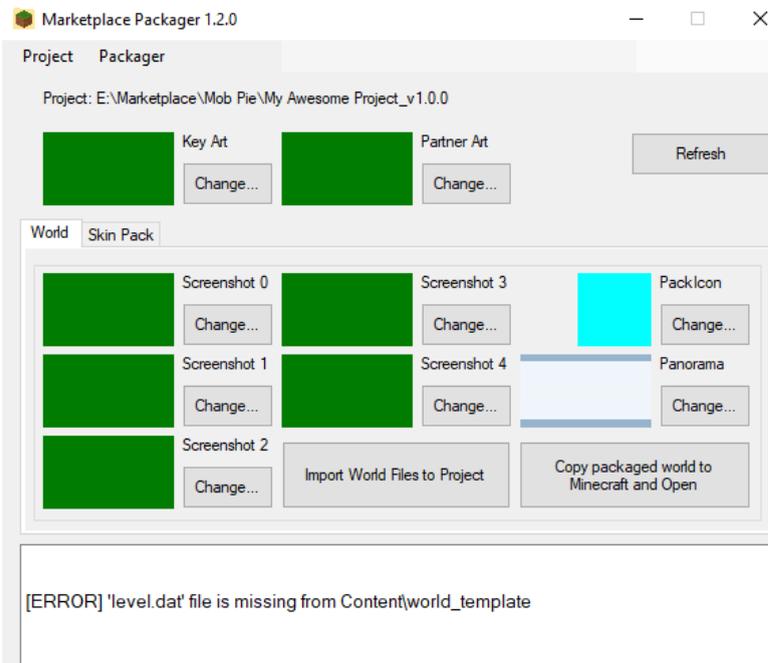
Under the world tab of a new project, click “Add world template”.

Because Map Packager can’t store any data in the project or Auger will complain, you will need to re-enter your project name. Make sure it’s exactly the same as you entered when creating the project.

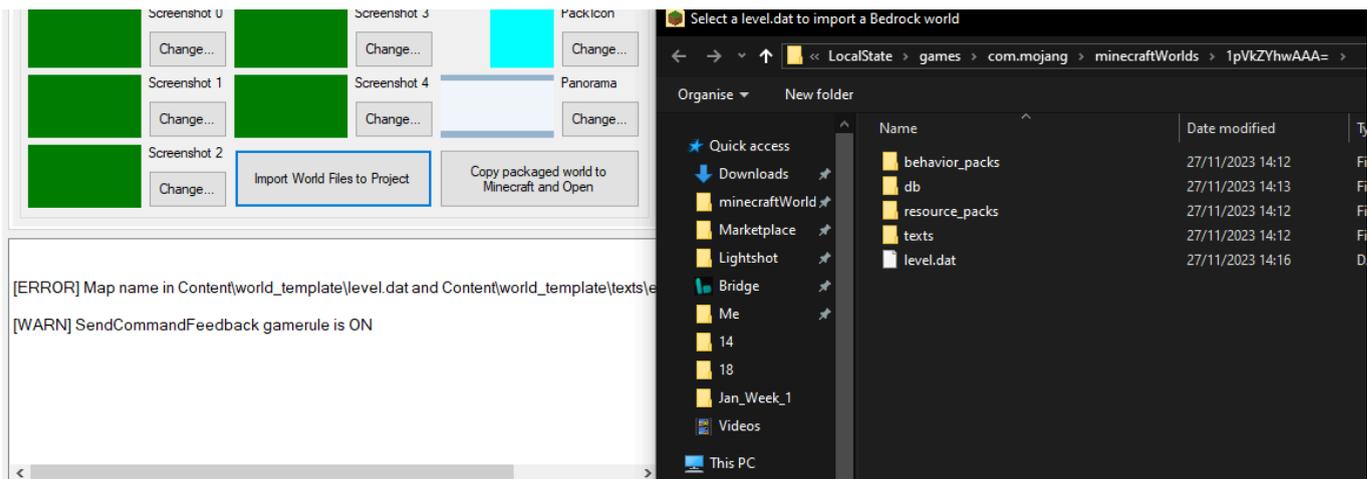
Enter a description. This is not your store description and most people just put a credit note here.

Click **Add** to continue.

Map Packager will now create a bunch of default files and folders in your project. These blank images are the size Auger expects so that you can edit and replace them manually if you wish.



You can see that we have an **error** because we don't have a world added yet. You can paste this manually in to the Content/world\_template folder, but it's recommended you just use the "Import World Files to Project" button and navigate to the level.dat file of your world.



In the example above, you can see we have an error and a warning. Map Packager has imported our world files, including any behavior and resource packs, and scanned the level.dat for potential issues.

**Errors:** Map Packager considers this a big issue that must be fixed before Auger will accept this world. In the above case, the map wasn't named the same thing as our project. I'll change the name of the world in Minecraft to "My Awesome Project" to fix this.

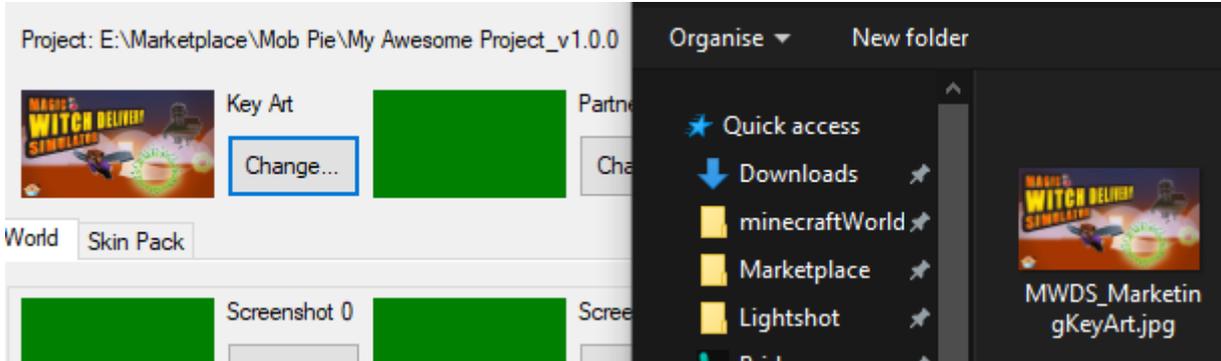
**Warnings:** These are things Map Packager has noticed that MIGHT be an issue. For example if your map contains command blocks, SendCommandFeedback should probably be off. Warnings might not be an issue though, it's just worth reviewing them. Common warnings are that the default gamemode is Creative or that daylight cycle is off.

Warnings are tuned to survival world style maps. They will warn you if mob spawning is off, or weather is off, or keep inventory is on, for example, so if you're making a puzzle map, you might get a lot of

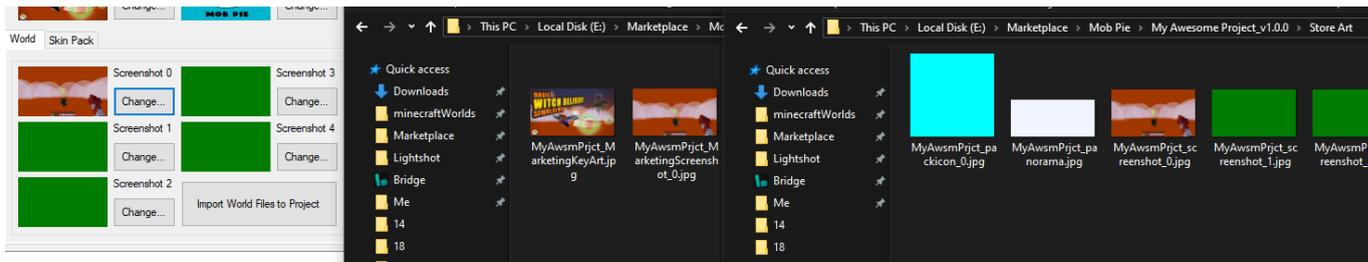
warnings. It's up to you to review your settings to make sure they make sense for your project. Map Packager just offers you information.

### Let's add our promotional art and store assets

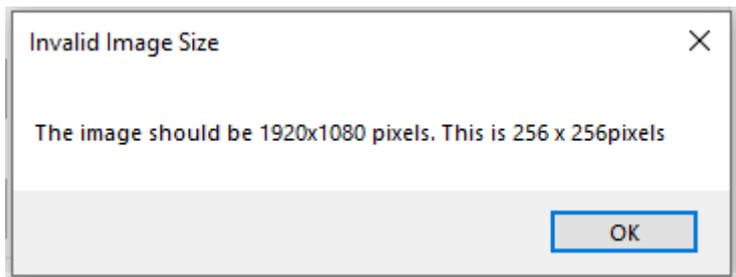
Each asset is represented in Map Packager by a preview picture, and a Change button. Clicking Change opens a file browser that you can use to select the **full size, 1080p version of your art**.



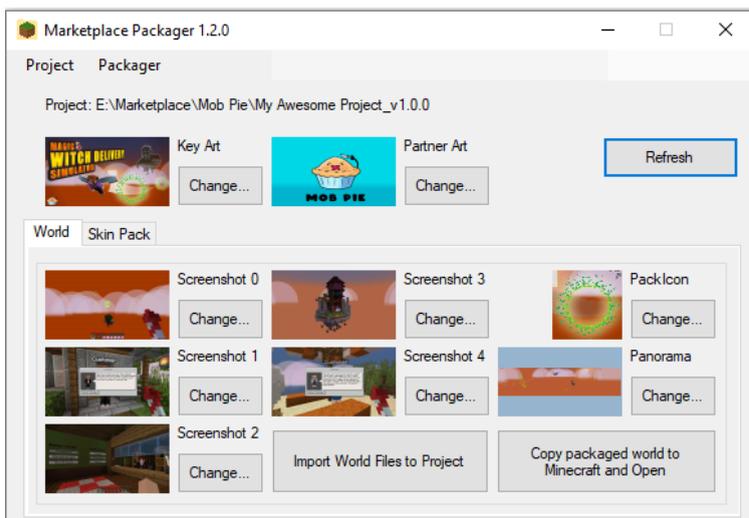
Map Packager will import and rename the image for you, as well as downscaling the image for the store art folder if applicable:



This saves you having to rename the image, resize it, or import it twice!



Add all of your marketing assets. If anything is the wrong size, Map Packager will warn you and not import the image.

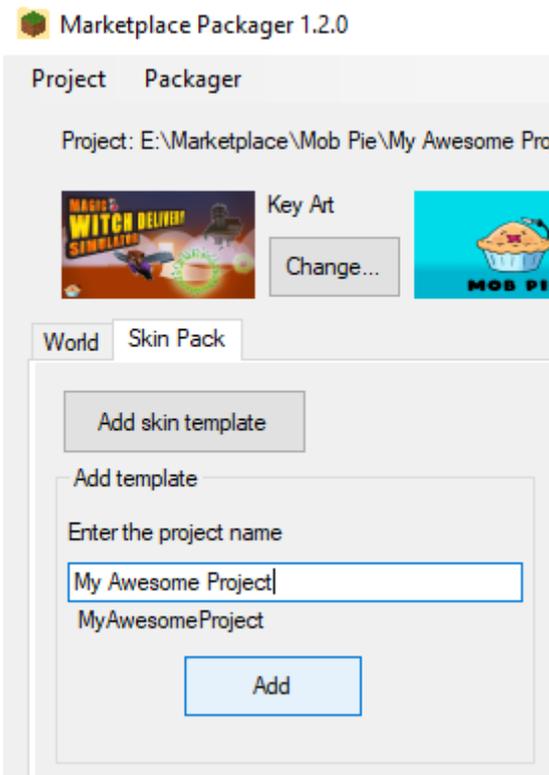


And we're done! Once we've cleared all the errors, and checked that the warnings don't apply to our map, we're ready to test and pack.

If you want to test a fresh copy of your packaged map (Recommended), click the "Copy packaged world to Minecraft and Open" button. You might want to check, for example, that the player spawn is set close

to where they log in, and that their inventory makes sense for your game - things Map Packager won't check for you.

## Add a Skin Pack to Your Project



Go to the Skin Pack tab of Map Packager and click "Add skin template". A new section will appear.

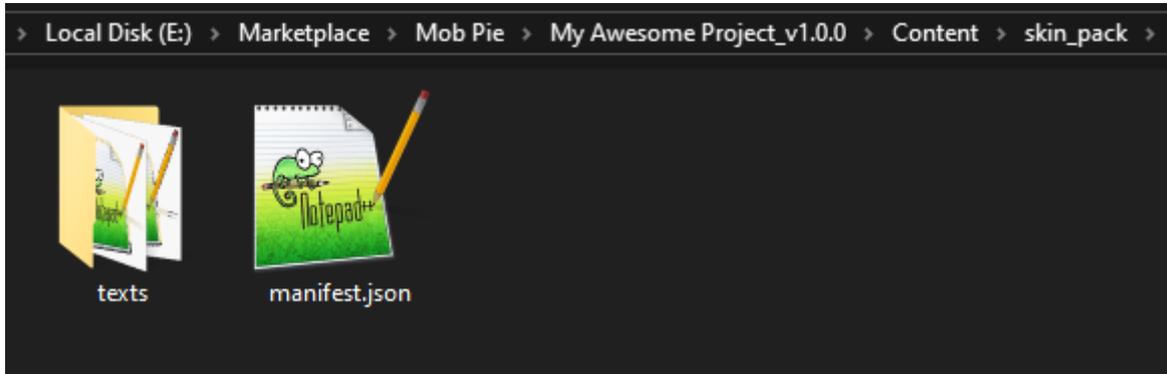
Because Map Packager can't store any data in the project or Auger will complain, you will need to re-enter your project name. Make sure it's exactly the same as you entered when creating the project.

Click **Add**



The Skin Pack tab will now look like this. You can see that there are instructions to add skin images to the project. There's also an error saying that skins.json is missing from the project. Don't worry, that will be created automatically in a later step.

Click the **Open Current Project Folder** button at the bottom of Map Packager. This will open the folder in Windows explorer. You can now navigate to Content/skin\_pack



You'll see a mostly empty folder. **This is where your skins go.**

## How you name your skins is very important.

Map packager uses the name of your skin file to decide:

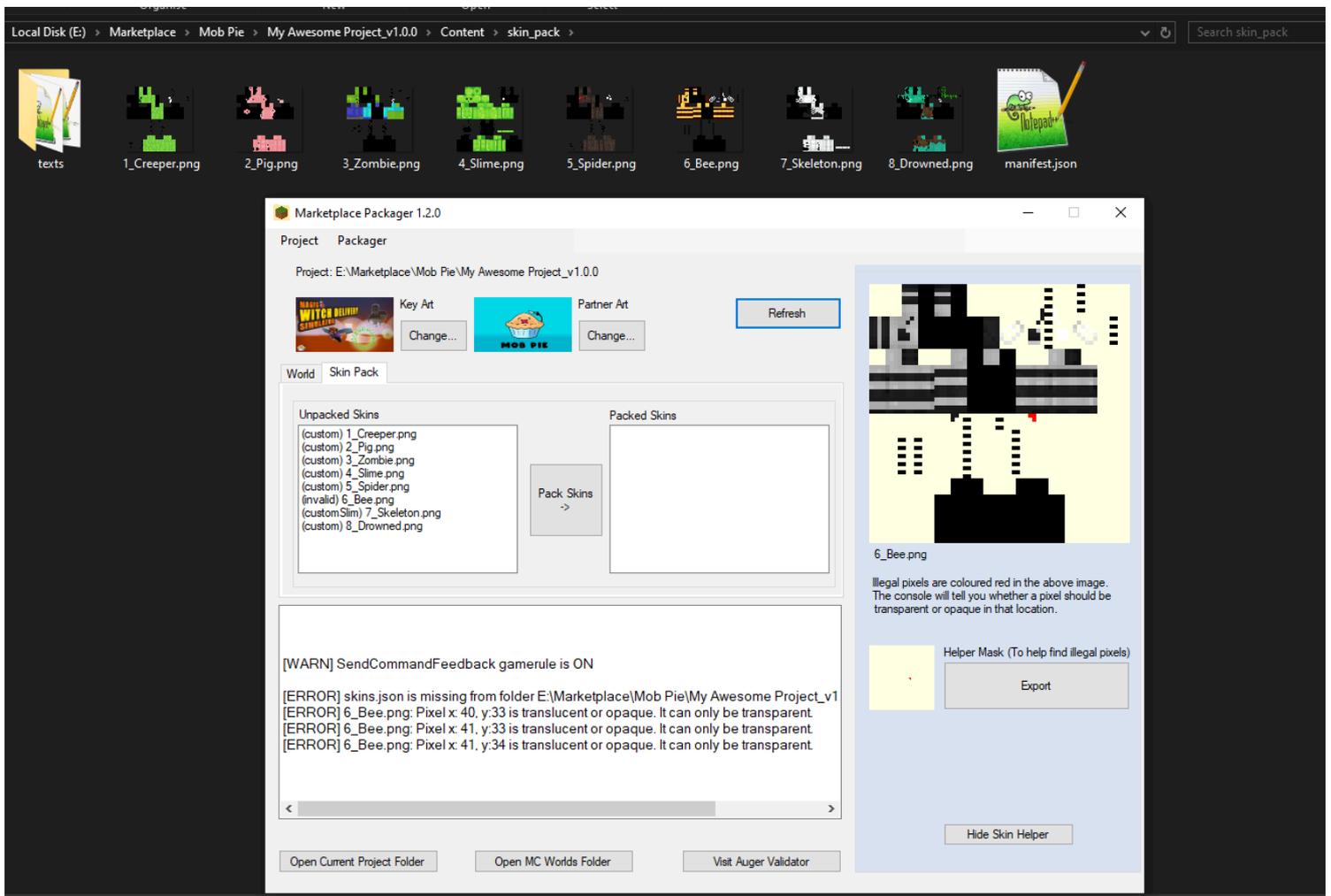
1. What to call the skin
2. What order to show the skins on the marketplace.

For example if your skin is called "Cyborg.png" then it will appear as "Cyborg" on the store.

**By default, Map Packager will order the skins arbitrarily!** This means that if you have the skins "Cyborg.png" and "Robot.png", Cyborg will appear first on the store and Robot second. To choose a custom order and make Robot appear first, you can add a number and an underscore to the start of your file name. "01\_Robot.png" and "02\_Cyborg", the store will display the skins Robot and Cyborg in that order.

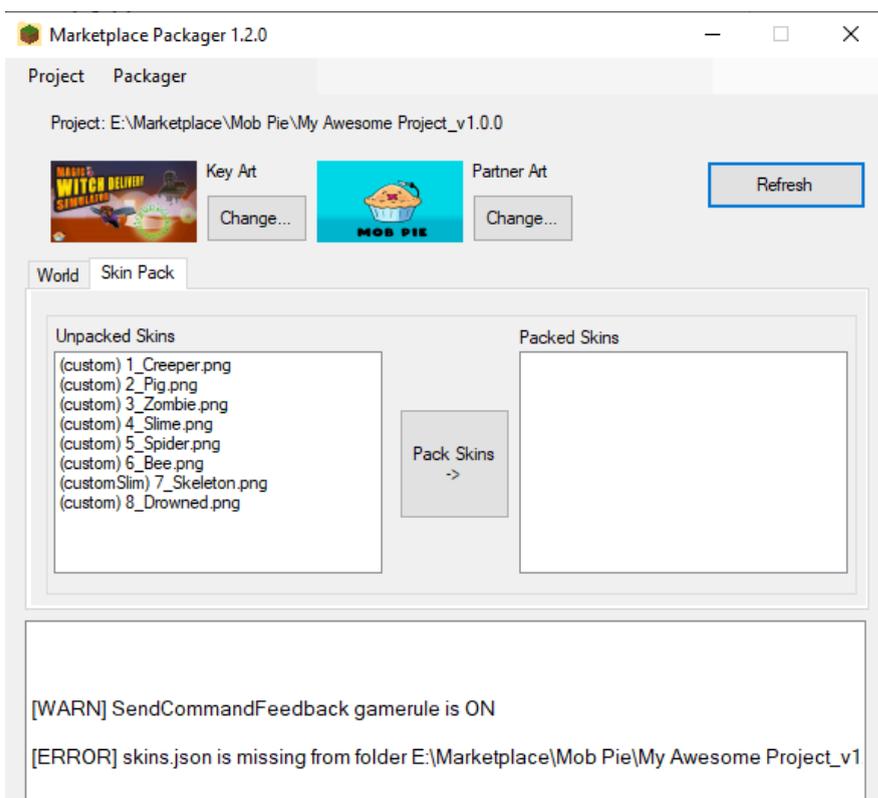
**Paste your skins as PNG files into the skin\_pack folder**

**Note: Currently Map Packager will tell you that HD skins are invalid. It will still pack them, but will complain at you.**



After copying my skins into the folder, I clicked **Refresh** in Map Packager. You can see that there are now 8 skins listed in **Unpacked Skins**. In front of the name in brackets you can see that it has detected that 6 skins are for the custom (Steve) model, one is for the customSlim (Alex) model, and one skin has **invalid pixels**. Looks like the artist in this example case has drawn some pixels that aren't allowed. The error messages show us the coordinates of the invalid pixels, and the **Skin Helper** has opened to show a black and white version of the skin with the offending pixels that need to be removed highlighted in red.

Let's go into our image editor and remove the pixels, **then Refresh again**.



Great, after refreshing, Skin Helper has closed itself and the skin is now listed as for the Custom model.

**We can now click the Pack Skins -> button**

All of the skins have been moved to the Packed Skins list! Success! The warning about the missing skins.json is now also gone. Map Packager has created the skins json with the skins ordered based on our number prefix, and update the language file to include the skin names from the file name.

Unpacked Skins

Drop your skin files into  
/Content/skin\_pack/  
Then click [Refresh]

Pack Skins  
->

Packed Skins

- 1\_Creeper.png
- 2\_Pig.png
- 3\_Zombie.png
- 4\_Slime.png
- 5\_Spider.png
- 6\_Bee.png
- 7\_Skeleton.png
- 8\_Drowned.png

[WARN] SendCommandFeedback gamerule is ON

**skins.json**

```
{
  "skins": [
    {
      "localization_name": "Skin1",
      "geometry": "geometry.humanoid.custom",
      "texture": "1_Creeper.png",
      "type": "paid"
    },
    {
      "localization_name": "Skin2",
      "geometry": "geometry.humanoid.custom",
      "texture": "2_Pig.png",
      "type": "paid"
    },
    {
      "localization_name": "Skin3",
      "geometry": "geometry.humanoid.custom",
      "texture": "3_Zombie.png",
      "type": "paid"
    },
  ],
}
```

en\_US.lang

```
skinpack.MyAwesomeProject=My Awesome Project
skin.MyAwesomeProject.Skin1=Creeper
skin.MyAwesomeProject.Skin2=Pig
skin.MyAwesomeProject.Skin3=Zombie
skin.MyAwesomeProject.Skin4=Slime
skin.MyAwesomeProject.Skin5=Spider
skin.MyAwesomeProject.Skin6=Bee
skin.MyAwesomeProject.Skin7=Skeleton
skin.MyAwesomeProject.Skin8=Drowned
```

**You're all done!**

You can now zip up your project and submit it on Auger.