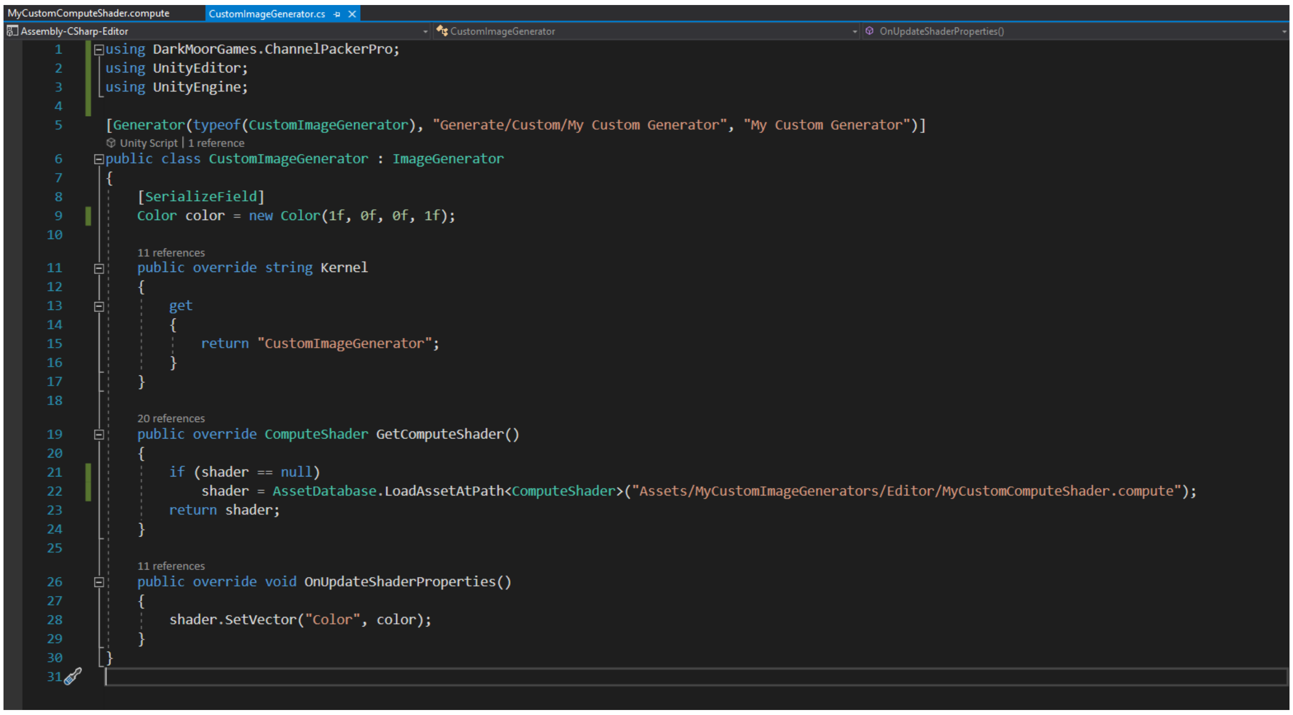
To create custom image generators create a script in an editor folder and make your class inherit from **ImageGenerator** which is in the **DarkMoorGames.ChannelPackerPro** namespace. Then add the **Generator** attribute which will allow channel packer pro to use your image generator and display it in the context menu.

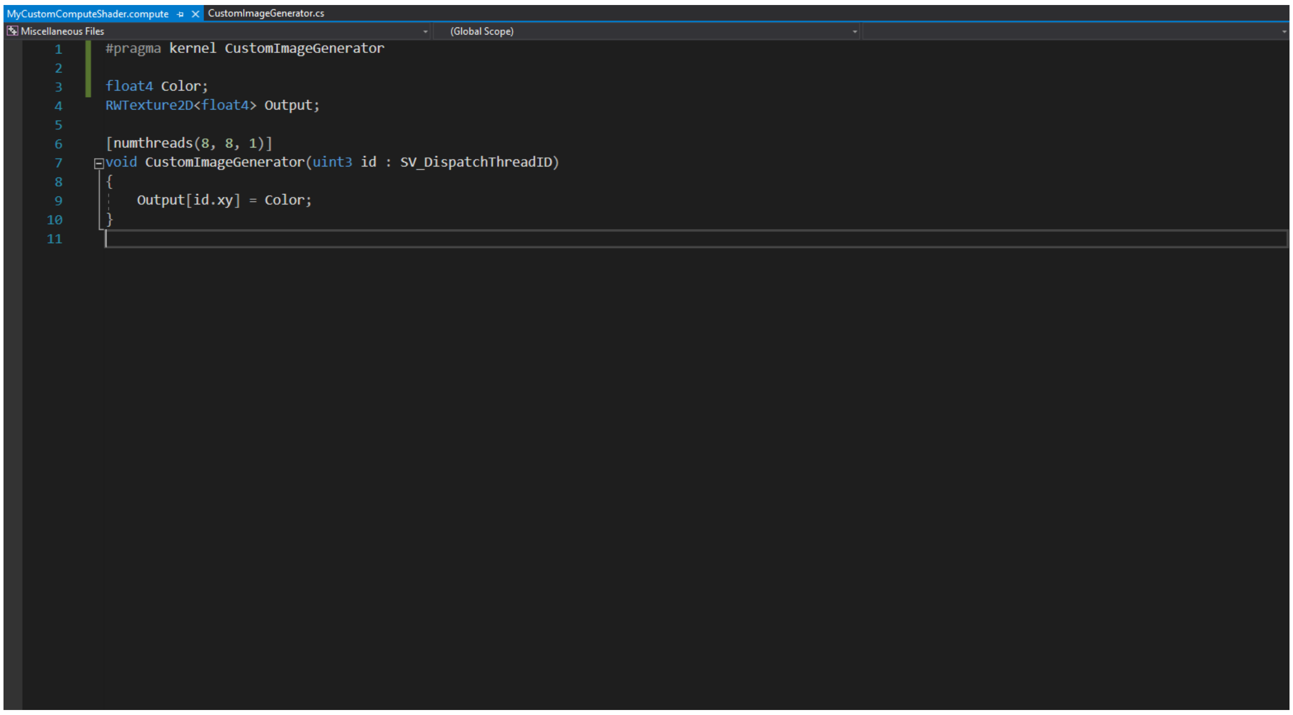
The first parameter of the **Generator** attribute is the type which is your class type and the second parameter is for the context menu and the third is its display name.

The **Kernel** property is the name of the compute shader kernel, the **GetComputeShader()** method should return the shader the generator uses, the **OnUpdateShaderProperties()** method gets called after any properties change like **Color** in the example below.



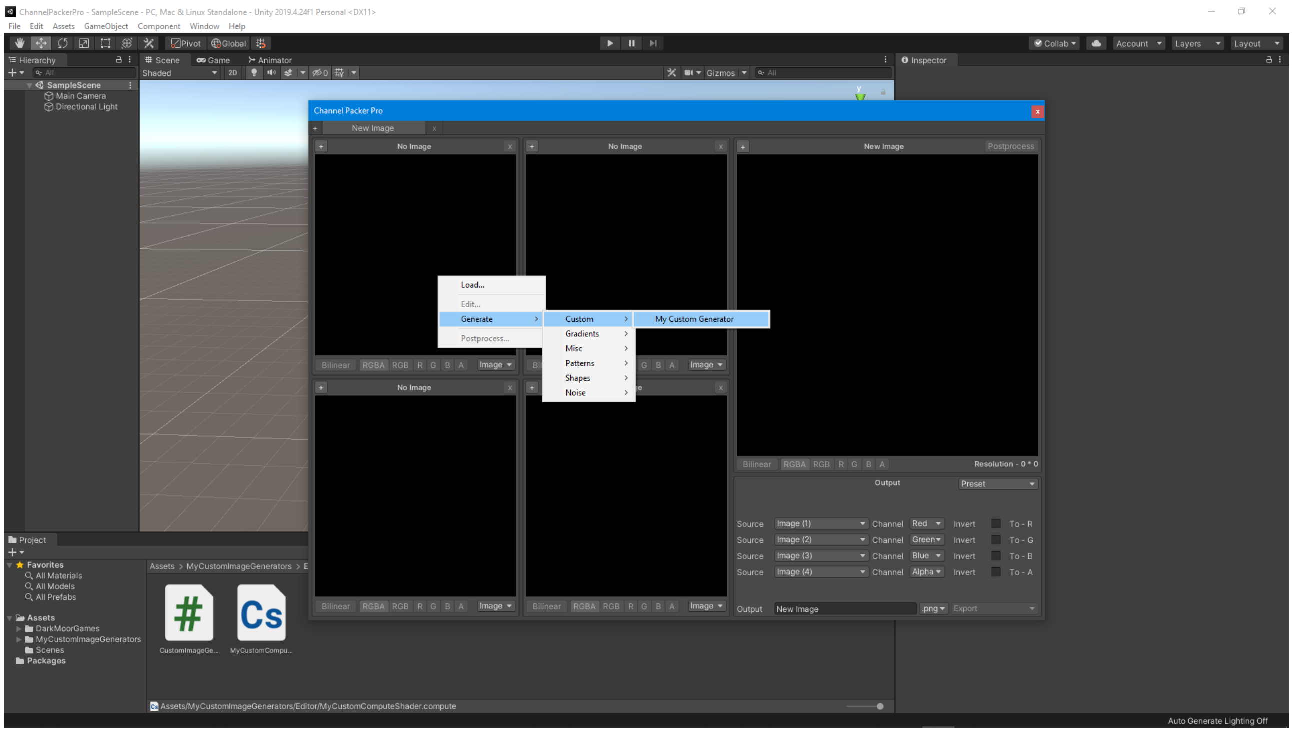
```csharp
using DarkMoorGames.ChannelPackerPro;
using UnityEditor;
using UnityEngine;

[Generator(typeof(CustomImageGenerator), "Generate/Custom/My Custom Generator", "My Custom Generator")]
public class CustomImageGenerator : ImageGenerator
{
    [SerializeField]
    Color color = new Color(1f, 0f, 0f, 1f);

    public override string Kernel
    {
        get
        {
            return "CustomImageGenerator";
        }
    }

    public override ComputeShader GetComputeShader()
    {
        if (shader == null)
            shader = AssetDatabase.LoadAssetAtPath<ComputeShader>("Assets/MyCustomImageGenerators/Editor/MyCustomComputeShader.compute");
        return shader;
    }

    public override void OnUpdateShaderProperties()
    {
        shader.SetVector("Color", color);
    }
}
```

Below is the compute shader that our custom image generator will use, notice the kernel name is the same as the property **Kernel** in our custom class above, the **Output** RWTexture is needed by all image generators.

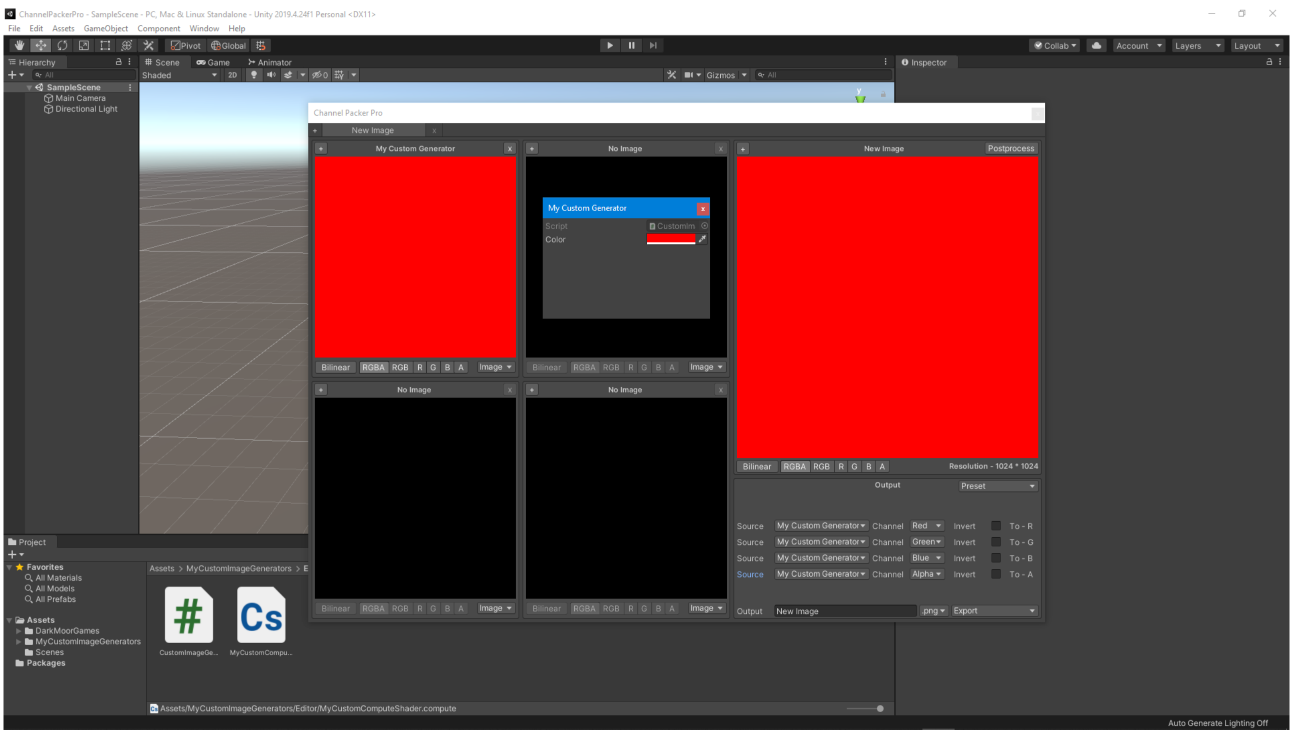


```hlsl
#pragma kernel CustomImageGenerator

float4 Color;
RWTexture2D<float4> Output;

[numthreads(8, 8, 1)]
void CustomImageGenerator(uint3 id : SV_DispatchThreadID)
{
    Output[id.xy] = Color;
}

```

Your custom image generator will now show in a context menu and can be used like in the image below.


ChannelPackerPro - SampleScene - PC, Mac & Linux Standalone - Unity 2019.4.24f1 Personal <DX11>
File Edit Assets GameObject Component Window Help
Hierarchy
SampleScene
Main Camera
Directional Light
Scene Game Animator
Inspector
Channel Packer Pro
New Image
No Image
No Image
New Image
Postprocess
Load...
Edit...
Generate
Postprocess...
Custom
Gradients
Misc
Patterns
Shapes
Noise
My Custom Generator
Bilinear RGBA RGB R G B A Image
Resolution - 0 * 0
Output
Preset
Source Image (1) Channel Red Invert To - R
Source Image (2) Channel Green Invert To - G
Source Image (3) Channel Blue Invert To - B
Source Image (4) Channel Alpha Invert To - A
Output New Image .png Export
Project
Favorites
All Materials
All Models
All Prefabs
Assets
DarkMoorGames
MyCustomImageGenerators
Scenes
Packages
Assets > MyCustomImageGenerators
CustomImageGe...
MyCustomCompu...
Assets/MyCustomImageGenerators/Editor/MyCustomComputeShader.compute
Auto Generate Lighting Off

Here you see after selecting the custom image generator a window with its display name show and has the **Color** property we created.



My Custom Generator
Script CustomIm
Color 🟥

Output Preset

Source	My Custom Generator	Channel	Red	Invert	☐	To - R
Source	My Custom Generator	Channel	Green	Invert	☐	To - G
Source	My Custom Generator	Channel	Blue	Invert	☐	To - B
Source	My Custom Generator	Channel	Alpha	Invert	☐	To - A

Output New Image .png Export

Assets/MyCustomImageGenerators/Editor/MyCustomComputeShader.compute