Manual

Mesh Materializer

Thank you for your purchase of **Mesh Materializer**.

Please consider writing a review or just rate the asset:

http://u3d.as/bfh

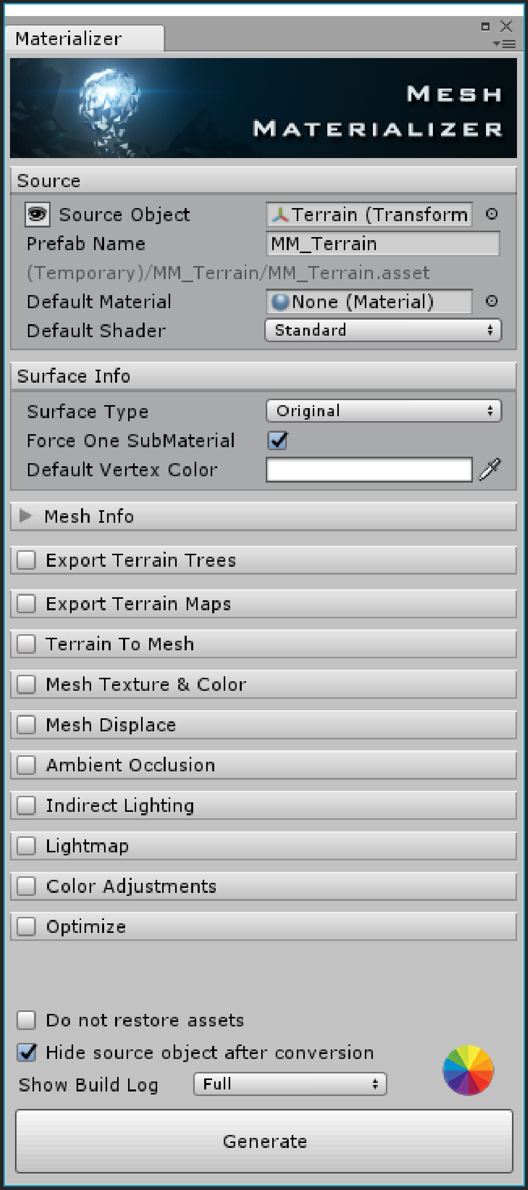
For any question or help use forum:

http://forum.unity3d.com/threads/mesh-materializer.293796/

Tutorial video playlist: https://goo.gl/3nSTAq

Mesh Materializer is accessible from menu Window/VacuumShaders/Mesh Materializer.

Hotkey: Ctrl + M


Materializer

Source

Source Object: Terrain (Transform)

Prefab Name: MM_Terrain

(Temporary)/MM_Terrain/MM_Terrain.asset

Default Material: None (Material)

Default Shader: Standard

Surface Info

Surface Type: Original

Force One SubMaterial: ✓

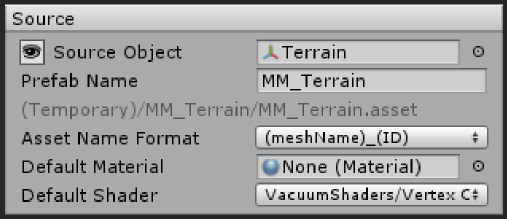
Default Vertex Color

☐ Do not restore assets

✓ Hide source object after conversion

Show Build Log: Full

Source

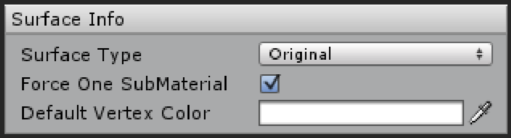

Source

Source Object	Terrain
Prefab Name	MM_Terrain
(Temporary)/MM_Terrain/MM_Terrain.asset	
Asset Name Format	(meshName)_(ID)
Default Material	None (Material)
Default Shader	VacuumShaders/Vertex C

Source Object – Object that will be converted (including its children in hierarchy).
Prefab Name – This is the name of prefab that will be created inside Assets/(Temporary)/ PREFAB NAME / folder.
Default Material – Generated prefab will use this material.
Default Shader – (If **Default Material** is not defined) after conversion Mesh Materializer will create material for assets using this shader. Material will have same name as **Prefab Name**.

- Button changes *Source Object's* visibility. If *Source Object* is child, than visibility is controlled by its parent object and this button is disabled.

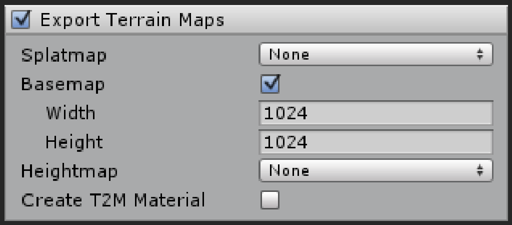
Surface Info


Surface Info

Surface Type	Original
Force One SubMaterial	✓
Default Vertex Color	

Surface Type – Should be saved original triangle type or it should be flattened. For flattened meshes vertex count = 3 * triangle count. If resultant mesh will have more than 65,000 vertices it will be split.
Force One SubMaterial – Generated mesh will combine all submaterials into one. Meshes with one submaterial require only 1 draw call for rendering.
Default Vertex Color – If during vertex color baking some data cannot be generated or is not provided enough info this color will be used.

Export Terrain Maps



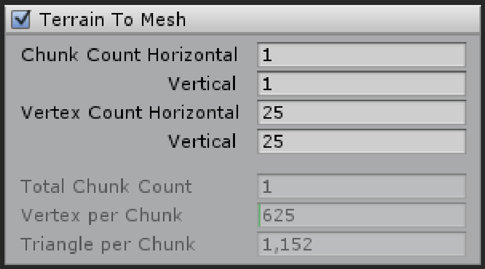
✓ Export Terrain Maps	
Splatmap	None
Basemap	✓
Width	1024
Height	1024
Heightmap	None
Create T2M Material	☐

Enables exporting terrain Splatmaps, Basemap and Heightmap textures. Files are saved in the same folder as the main prefab.

- Splatmap –Uncompressed RGBA texture used by Unity Terrain engine for blending paint textures. One splatmap can blend 4 paint textures. Splatmaps can be exported in PNG and TGA (requires Encode To TGA asset) formats.
- Basemap – All paint textures used by Unity Terrain engine are baked into one final texture. Diffuse and Normal textures are exported separately. Requires device with RenderTextures support.
- Heightmap – Grayscale texture with terrain height data. Exported texture size depends on *Heightmap Resolution* defined inside source *Terrain Settings.*
    - Original – Exports original heightmap.
    - Remap – Before exporting terrain heightmap data is remapped to be inside [0, 1] range.

Create T2M Material – Generates T2M material for converted terrain. Available only if splatmap or basemap exporters are enable.

Terrain To Mesh

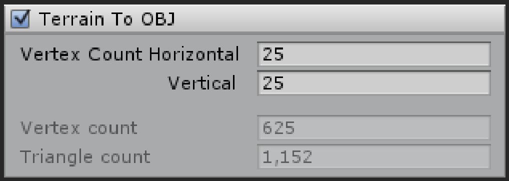


✓ Terrain To Mesh	
Chunk Count Horizontal	1
Vertical	1
Vertex Count Horizontal	25
Vertical	25
Total Chunk Count	1
Vertex per Chunk	625
Triangle per Chunk	1,152

Converts terrain to mesh.

Chunk Count Horizontal/Vertical – Instead of converting source terrain into one mesh, it can be divided into multiple chunks.
Vertex Count Horizontal/Vertical – Chunk vertex count. Each chunk may have max 65,000 vertices.

Terrain To OBJ



✓ Terrain To OBJ

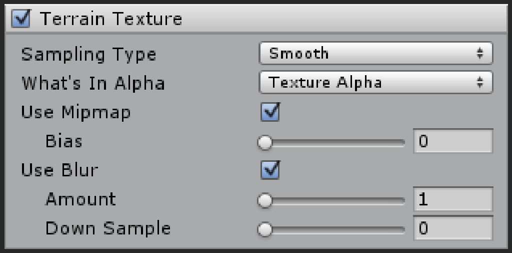
Vertex Count Horizontal	25
Vertical	25
Vertex count	625
Triangle count	1,152

Converts terrain into .obj file format. Can be imported into any 3d modeling software for additional editing.

Vertex count horizontal/vertical – OBJ file has no limit on vertex count, if mesh has more than 65.000 vertices it will be split by Unity automatically. UV and normals are generated by default, tangents are calculated by Unity after file importing. OBJ file is saved inside same folder as the main asset, in OBJ subfolder.

OBJ file does not save vertex color and flat surface info.

Terrain Textures



✓ Terrain Texture

Sampling Type	Smooth
What's In Alpha	Texture Alpha
Use Mipmap	✓
Bias	0
Use Blur	✓
Amount	1
Down Sample	0

Bakes terrain textures into generated mesh vertex color.

Sampling Type (available only if surface type is Flat)

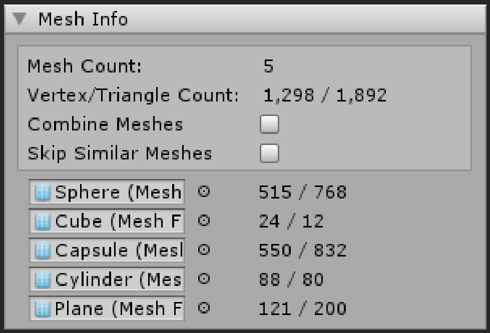
- Smooth
- Flat Hard
- Flat Smooth
- Flat Smoother

What's In Alpha - Defines what is saved inside vertex color's alpha channel.

Use Mipmap – Reads texture data from its mipmap, cheap and fast way for achieving blur effect.

Use Blur - GPU accelerated texture blur effect using Gaussian filters. Requires RenderTextures support.

Mesh Info



Mesh Count: 5
Vertex/Triangle Count: 1,298 / 1,892
Combine Meshes ☐
Skip Similar Meshes ☐

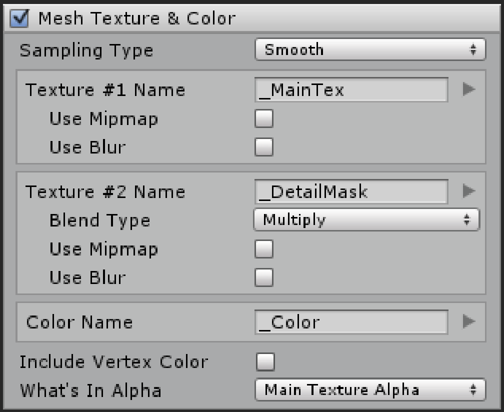
Sphere (Mesh	515 / 768
Cube (Mesh F	24 / 12
Capsule (Mesl	550 / 832
Cylinder (Mes	88 / 80
Plane (Mesh F	121 / 200

Displays simple and skinned mesh info (vertex and triangle count).

Combine Meshes – (Available only with simple meshes) can combine meshes into one and saves one mesh asset.

Skip Similar Meshes – If *Source Object* contains multiple same meshes (mesh name defines *similarity*) there is no need to convert and save them all. It's enough to convert only one of them and in final prefab it will be used. But in some cases it is necessary to convert all meshes in the *Source Object* – in this case this option must be turned off.

Mesh Texture & Color



Sampling Type: Smooth
Texture #1 Name: _MainTex
Use Mipmap ☐
Use Blur ☐
Texture #2 Name: _DetailMask
Blend Type: Multiply
Use Mipmap ☐
Use Blur ☐
Color Name: _Color
Include Vertex Color ☐
What's In Alpha: Main Texture Alpha

Bakes texture and color into generated mesh vertex color.
Texture and Color data are taken from material used by mesh.

Sampling Type (available only if surface type is Flat)

- Smooth
- Flat Hard
- Flat Smooth
- Flat Smoother

Texture #1 Name - Texture parameter inside material from where texture data will be read.
▶ Pick-up button displays texture parameters that are available inside material.
To bake texture it must be readable. Check "Read/Write Enabled" inside texture import settings.
Unity readable texture formats are - ARGB32, RGBA32, BGRA32, RGB24, Alpha8 and DXT.

Use Mipmap – Reads texture data from its mipmap, cheap and fast way for achieving blur effect.

Use Blur - GPU accelerated texture blur effect using Gaussian filters. Requires RenderTextures support.

Texture #2 Name (available only if Texture #1 is active) – Mesh Materializer can bake two textures blending them by **Blend Type** parameter.

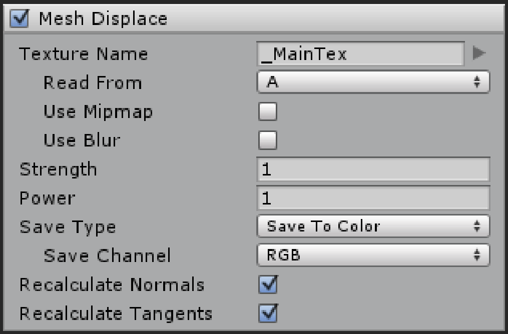
Color Name - Color parameter inside material from where color data will be read.
▶ Pick-up button displays color parameters that are available inside material.

Include Vertex Color – If source mesh has vertex color it can be saved within generated mesh.

*Final baked vertex color is calculated by: Texutre * Color * VertexColor.*
*If data cannot be baked **Default Vertex Color** will be used.*

What's In Alpha - Defines what is saved inside vertex color's alpha channel.

Mesh Displace



Mesh Displace ✓	
Texture Name	_MainTex
Read From	A
Use Mipmap	☐
Use Blur	☐
Strength	1
Power	1
Save Type	Save To Color
Save Channel	RGB
Recalculate Normals	✓
Recalculate Tangents	✓

Adds displace baking. Saves data inside vertex color or can modify final mesh vertices position.

Texture Name - Texture parameter inside material from where displace data will be read.
▶ Pick-up button displays texture parameters that are available in material.
To bake texture it should be readable. Check "Read/Write Enabled" inside texture import settings.
Unity readable texture formats are - ARGB32, RGBA32, BGRA32, RGB24, Alpha8 and DXT.

Read From – Displace data can be read from any channel of texture, or from its grayscale value.

Use Mipmap – Reads texture data from its mipmap, cheap and fast way for achieving blur effect.

Use Blur - GPU accelerated texture blur effect using Gaussian filters. Requires RenderTextures support.

Strength – Displace strength.

Power – Mathematical “power” of displace value.

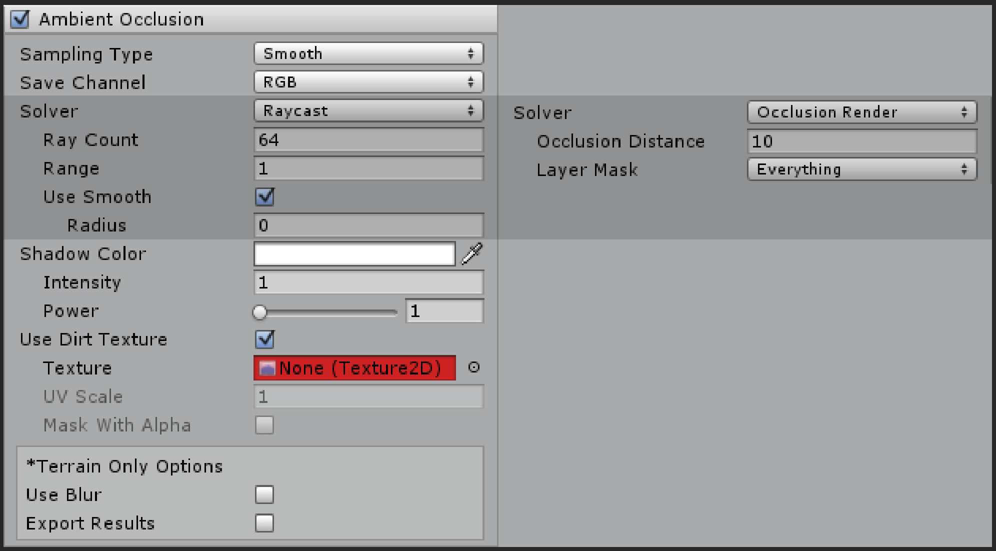
Save Type – How should be saved displace

- Displace Vertex
- Save To Color – Instead of displacing vertices, displace data will be saved inside color (RGB or Alpha channel). Color value will be multiplied to already baked color data.
- Displace vertex and save to color.

Recalculate Normals – After displace vertex normals need recalculation.

Recalculate Tangents – After displace tangent buffer needs recalculation.

Ambient Occlusion



✓ Ambient Occlusion	
Sampling Type	Smooth
Save Channel	RGB
Solver	Raycast
Ray Count	64
Range	1
Use Smooth	✓
Radius	0
Shadow Color	⬜
Intensity	1
Power	1
Use Dirt Texture	✓
Texture	None (Texture2D)
UV Scale	1
Mask With Alpha	☐
*Terrain Only Options	
Use Blur	☐
Export Results	☐

Solver	Occlusion Render
Occlusion Distance	10
Layer Mask	Everything

Generates ambient occlusion and saves results in vertex color.

Sampling Type (available only if surface type is Flat) – Smooth or Flat

Save Channel – Where in vertex color is saved generated results, Inside RGB or Alpha.

Two ambient occlusion solvers are available: Raycast and Occlusion Render.

Raycast – Generates AO by throwing multiple rays from each vertex. Solver runs on any device. Solver cannot interact with scene’s other meshes, only with source object.

Ray Count – More rays produce better quality but need more time to calculate final value.

Range – Ray length

Use Smooth – Smooth results with neighbor vertices.

Radius – Max distance to neighbor vertex.

Occlusion Render - For AO calculating object is rendered from each vertex position and then determined average value. Solver produces smoothest result (as infinity ray count in Raycast solver), is faster on high poly meshes and can interact with other scene meshes. Requires device with RenderTextures support.

Occlusion Distance – Visibility distance of vertex.

Layer mask – Defines object layers that interact with AO solver. If it is set to 'Nothing' – nothing will be rendered, even selected object.

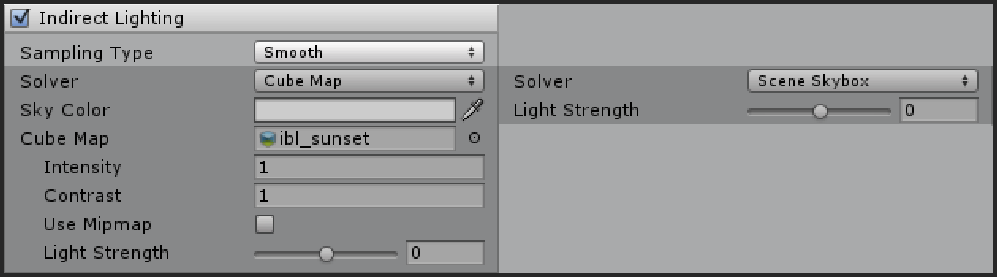
Shadow Color – AO color if it is saved inside RGB channel.

Intensity / Power – AO result intensity and power.

Use Dirt Texture – Final AO color will be multiplied by dirt texture color(RGB) and its alpha(if **Mask With Alpha** is enabled). Check "Read/Write Enabled" inside texture import settings.
Unity readable texture formats are - ARGB32, RGBA32, BGRA32, RGB24, Alpha8 and DXT.

If ambient occlusion is generated for terrain, results can be blurred and exported as texture. Texture is saved inside same folder as main prefab and its dimension equals exported terrain vertex count horizontal and vertical.

Indirect Lighting



✓ Indirect Lighting	
Sampling Type	Smooth
Solver	Cube Map
Sky Color	
Cube Map	ibl_sunset
Intensity	1
Contrast	1
Use Mipmap	☐
Light Strength	0

Solver	Scene Skybox
Light Strength	0

Generates indirect lighting and saves results in vertex color.

Sampling Type (available only if surface type is Flat) – Smooth or Flat

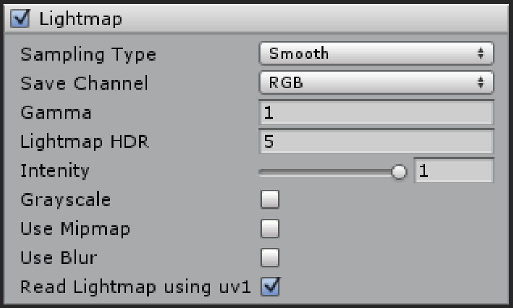
Two indirect lighting solvers are available: CubeMap and Scene Skybox.

CubeMap – Generates indirect lighting from simple cubemap. Solver is very fast and runs on any device.

Cubemap must be readable. Unity readable texture formats are - ARGB32, RGBA32, BGRA32, RGB24, Alpha8 and DXT.

Scene Skybox – Indirect lighting is generated from scene's active skybox. Solver works with all types of skybox shaders. Requires device with RenderTextures support.

Lightmap



✓ Lightmap	
Sampling Type	Smooth
Save Channel	RGB
Gamma	1
Lightmap HDR	5
Intenity	1
Grayscale	☐
Use Mipmap	☐
Use Blur	☐
Read Lightmap using uv1	✓

Bakes lightmap to vertex color (if mesh is lightmap).

For lightmap baking texture must be readable.
Unity readable texture formats are - ARGB32, RGBA32, BGRA32, RGB24, Alpha8 and DXT.

Sampling Type (enabled only if surface type is Flat)

- Smooth
- Flat Hard
- Flat Smooth
- Flat Smoother

Save Channel – where is saved generated lightmap, Inside RGB or Alpha.
Color value will be multiplied with already baked color data.

Gamma/HDR/Intensity –

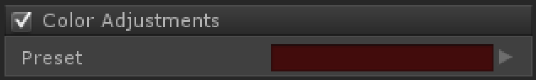
Grayscale – Instead of colored lightmap will be saved its grayscale values.

Use Mipmap – Reads texture data from its mipmap, cheap and fast way for achieving blur effect.

Use Blur - GPU accelerated texture blur effect using Gaussian filters. Requires RenderTextures support.

Read Lightmap using uv1 - By default Lightmap is read using uv2, but it can also be read by uv1.

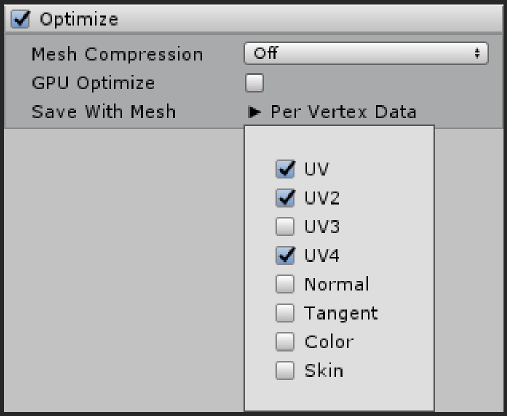
Color Adjustments


☑ Color Adjustments
Preset 🟥

Applies color adjustments to the generated vertex colors.

Preset - Color adjustments preset name. Pick-up button displays available color adjustment presets.

Optimize

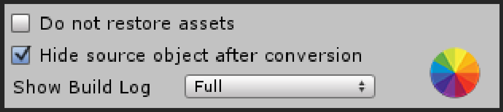

☑ Optimize
Mesh Compression: Off
GPU Optimize: ☐
Save With Mesh: ► Per Vertex Data
☑ UV
☑ UV2
☐ UV3
☑ UV4
☐ Normal
☐ Tangent
☐ Color
☐ Skin

Mesh Compression - Compressing meshes saves space in the built game, but more compression introduces more artifacts in vertex data.

GPU Optimize - Optimizes the mesh for GPU access.

Save With Mesh – Excluding per-vertex data reduce file size.

Misc


☐ Do not restore assets
☑ Hide source object after conversion
Show Build Log: Full

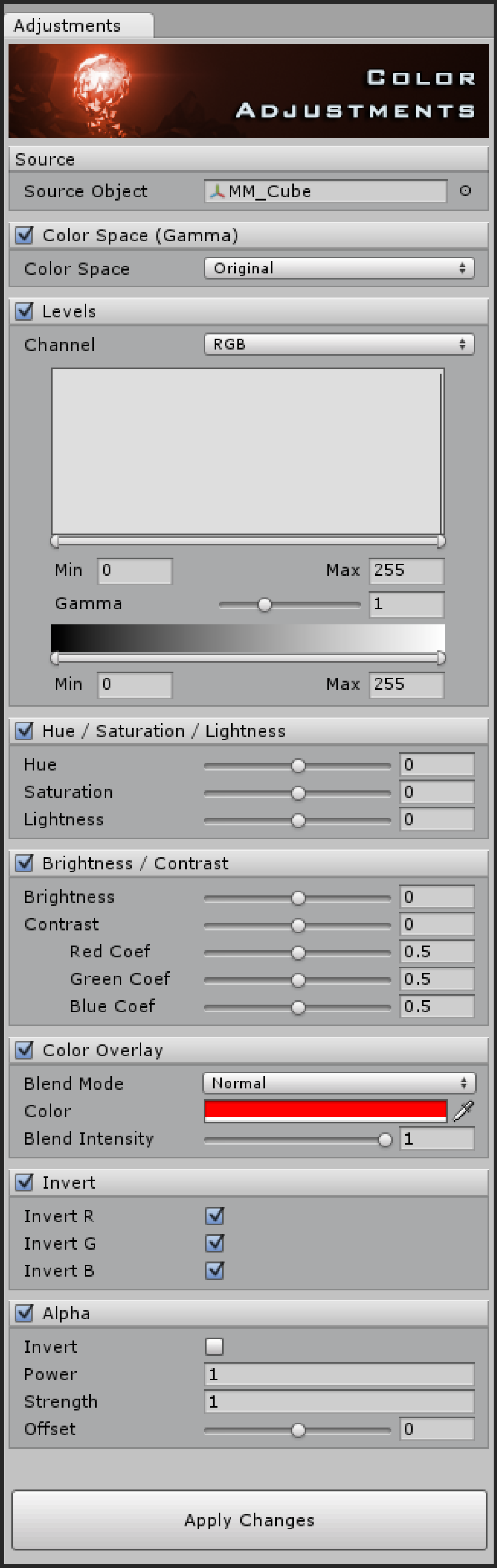
Do not restore assets – If baked textures and meshes are not readable or have unsupported file formats, before conversion Mesh Materializer changes their import settings and forces Unity to reimport them. After conversion files are reimported again with their original import settings. All these steps require much more time then conversion itself. Checking this option speeds up total conversion time minimum twice.

Hide source object after conversion – After conversion new prefab is instantiated in the scene in the same position as source object. It’s helpful to hide source object in this case.

Source object’s visibility can be controlled by 👁 icon.

Show Build Log – Outputs useful info about conversion.

Color Adjustments is accessible from menu Window/VacuumShaders/Color Adjustments or by clicking button inside Mesh Materializer window.



Adjustments

Source

Source Object: MM_Cube

☑ Color Space (Gamma)

Color Space: Original

☑ Levels

Channel: RGB

Min 0 Max 255

Gamma 1

Min 0 Max 255

☑ Hue / Saturation / Lightness

Hue	0
Saturation	0
Lightness	0

☑ Brightness / Contrast

Brightness	0
Contrast	0
Red Coef	0.5
Green Coef	0.5
Blue Coef	0.5

☑ Color Overlay

Blend Mode	Normal
Color	🟥
Blend Intensity	1

☑ Invert

Invert R	☑
Invert G	☑
Invert B	☑

☑ Alpha

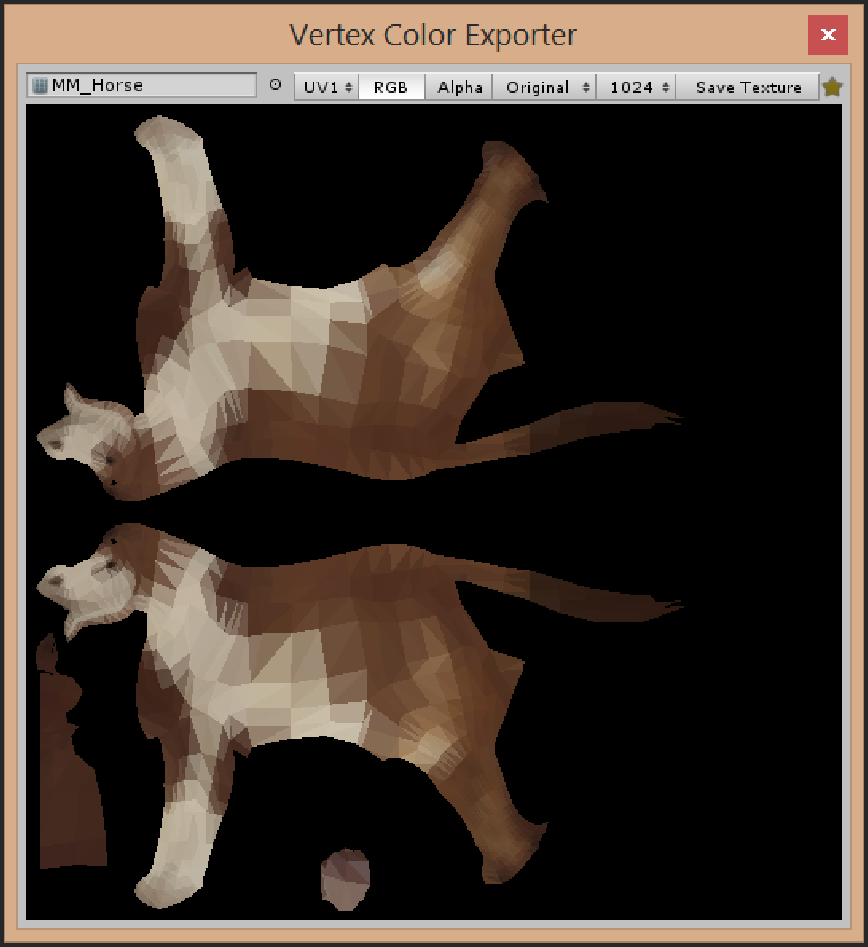
Invert	☐
Power	1
Strength	1
Offset	0

Apply Changes

Color adjustments are applied only to the selected mesh vertex colors and only while CA window is active. If CA window losses focus selected mesh will restore its original vertex color.

Apply changes button will apply color adjustments changes to the mesh vertex color (without undo).

Vertex Color Exporter is accessible from menu Window/VacuumShaders/Vertex Color Exporter



Exports mesh vertex color as texture in png format.

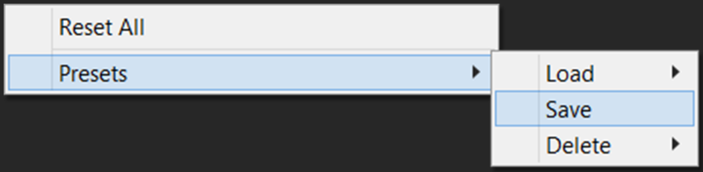
Can export only RGB and Alpha channels, or both together.

Exporting texture with antialiasing enabled may crash Unity on some hardware (Unity bug).

Working with presets

Mesh Materializer and Color Adjustments settings can be saved as presets.

Presets are accessible from contex (right click) menu for each window.


Reset All
Presets
Load
Save
Delete