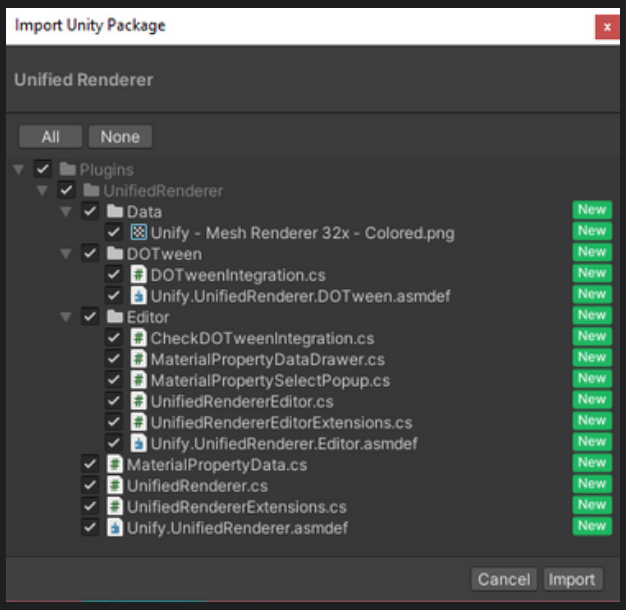
Contact: *contact@lukakldiashvili.com* **GitHub:** *https://github.com/lukakldiashvili/Unified-Renderer-Unity*

Unified Renderer

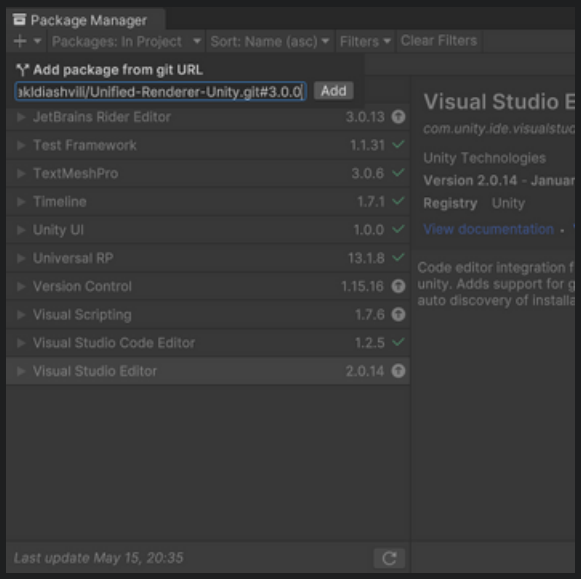
Setup & Guide

Step 1.1: Installation

Asset Store / Unity Package



Git URL

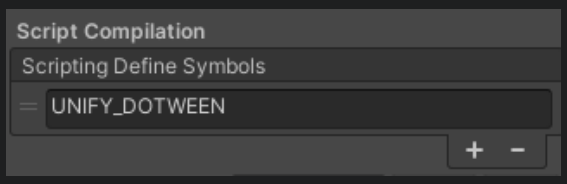


NOTE: git URL can by found on the GitHub page

Step 1.2: (Automatic) Integrations

Unified Renderer has optional third party integrations such as: DOTween. Such integrations will be automatically enabled/disabled, based on their presence in the project.

Define Symbols are used for the behaviors mentioned.



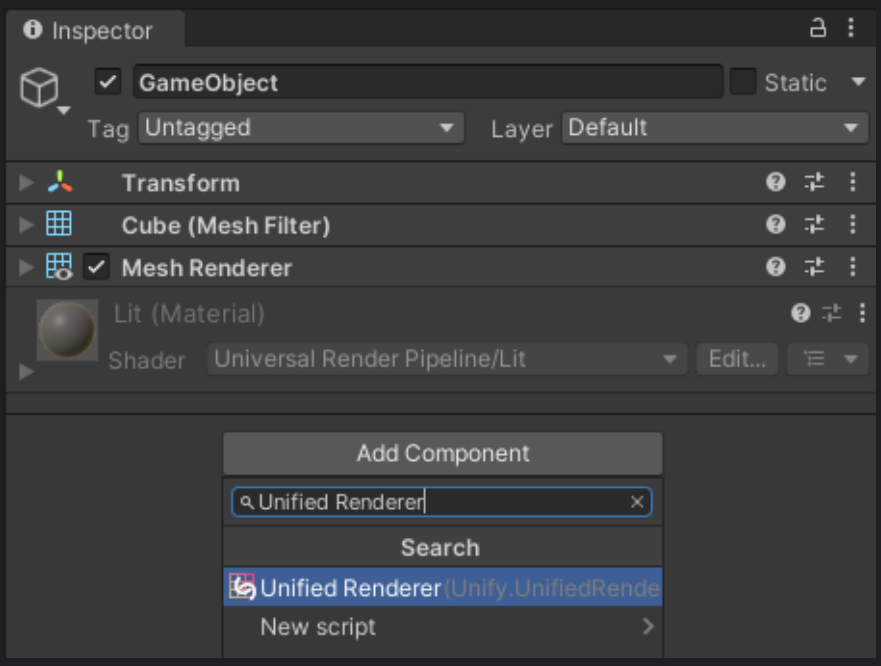
You will be notified when these actions occur:

Unified Renderer: Automatically enabled DOTween Integration

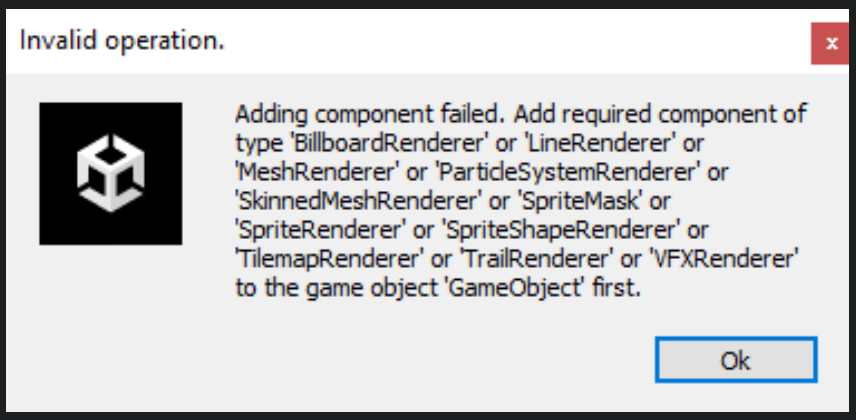
Unified Renderer: Automatically removed DOTween Integration

Step 2.1: Adding The Component

Add Unified Renderer component on any GameObject, which already has supported Renderer type attached (eg: MeshRenderer, SkinnedMeshRenderer, etc)



If presented with following message, please add one of the mentioned components, before adding Unified Renderer component.

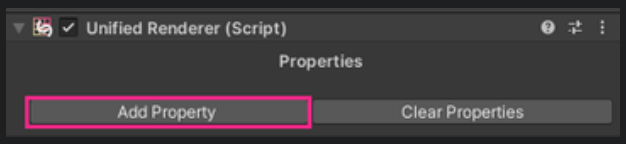


Once Unified Renderer has been successfully attached to a GameObject, you can start playing with material properties.

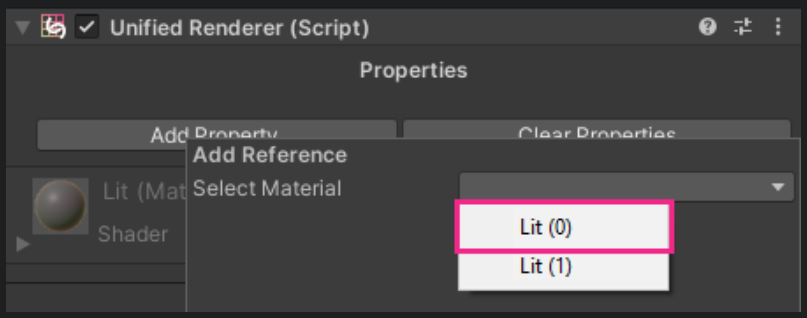
Step 2.2: Using The Component

Replicate following steps as an example:

1. Press 'Add Property' button

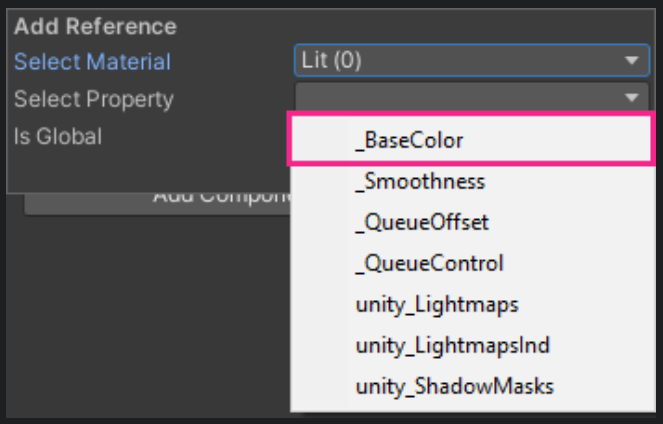


2. Select material you want to modify



3. Select property you want to override

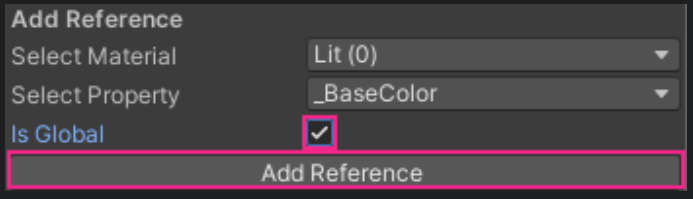
NOTE: your list of properties may be quite larger, that is because default materials have a lot of properties. For simplicity, we are using custom shader with 2 properties.



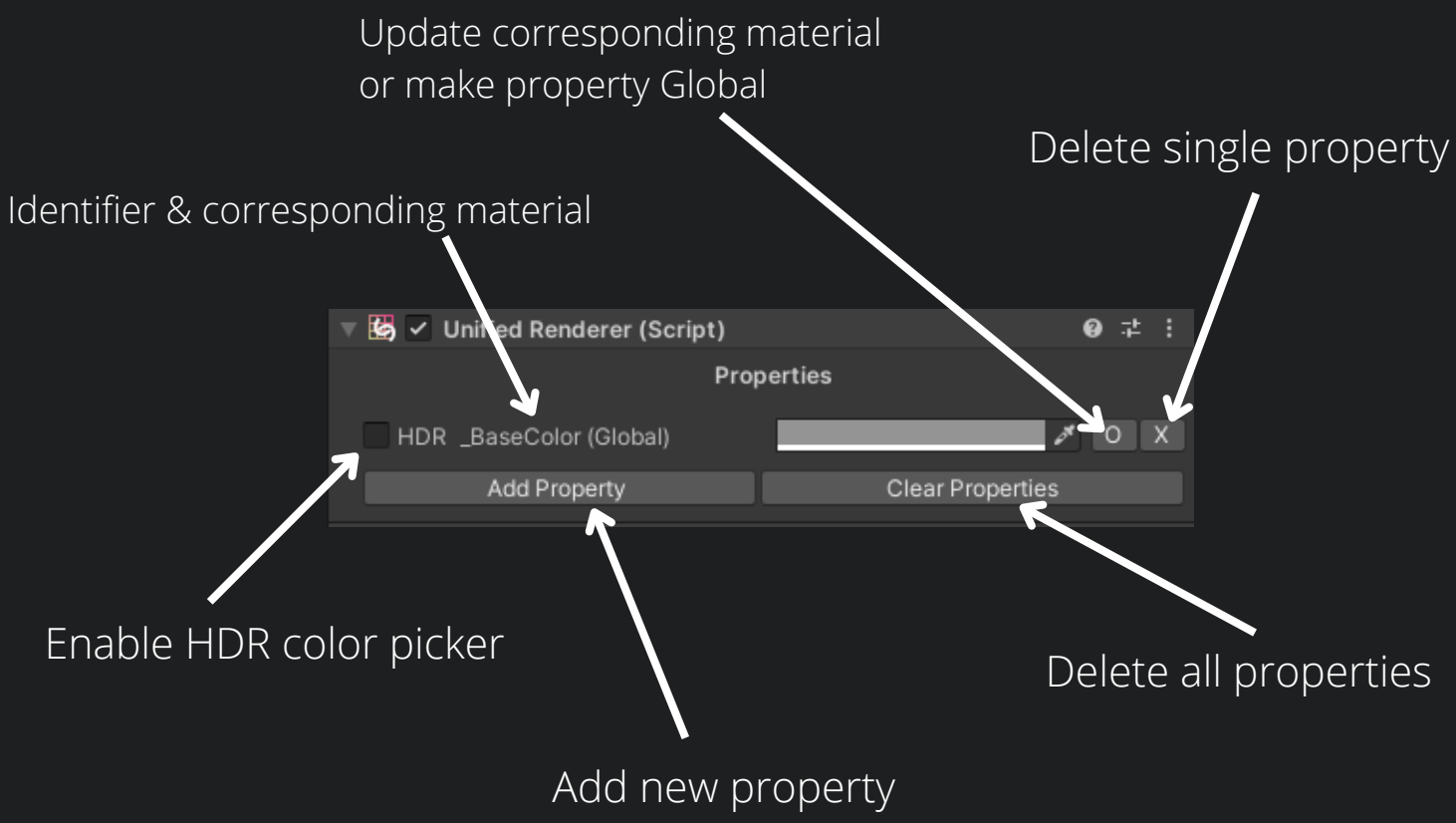
4. Is Global Property

'Is Global' - means that selected property will be overridden for every material present on the renderer, which contain selected property.

5. Finish by pressing 'Add Reference'



Step 3.1: Understanding The Component



Step 3.2: Understanding The API

Adds new property dynamically and sets default value, or value given by non-generic version of the method.

```
unifiedRenderer.AddProperty<Color>("_BaseColor");
```

Type of property we are setting (must be same as material property's type)

Property identifier

```
unifiedRenderer.AddProperty("_BaseColor", Color.white);
```

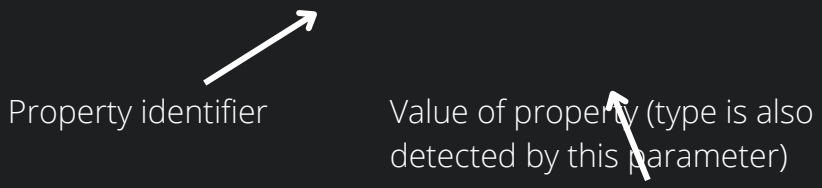
Property identifier

Value of property (type is also detected by this parameter)

returns: 'MaterialPropertyData' which contains data about property that was generated.

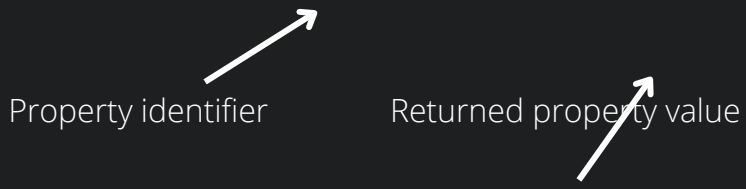
Sets new value to already existing property. If not present, adds correct one automatically.

```
unifiedRenderer.TrySetPropertyValue("_BaseColor", Color.red);
```



Gets value of already existing property.

```
unifiedRenderer.TryGetPropertyValue("_BaseColor", out Color outColor);
```



returns: 'bool' which means if method worked successfully.

Step 4: The Feedback

Thank you for downloading Unified Renderer and reading this manual.

Unified Renderer is open-sourced and open to contribution. Please send me the feedback to the contact email provided below or on the github page.

Contact: *contact@lukakldiashvili.com* **GitHub:** *https://github.com/lukakldiashvili/Unified-Renderer-Unity*