User Guide

Universal Settings

by lincolncpp

Asset for Unity

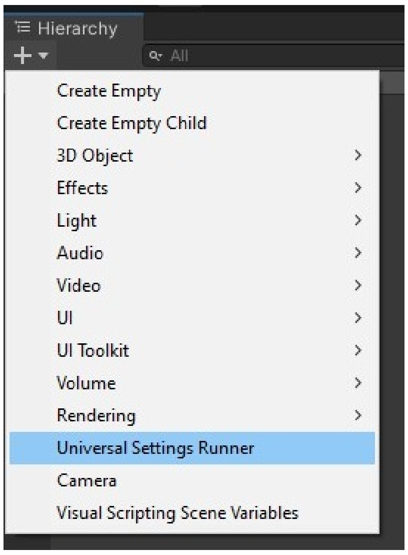
2023

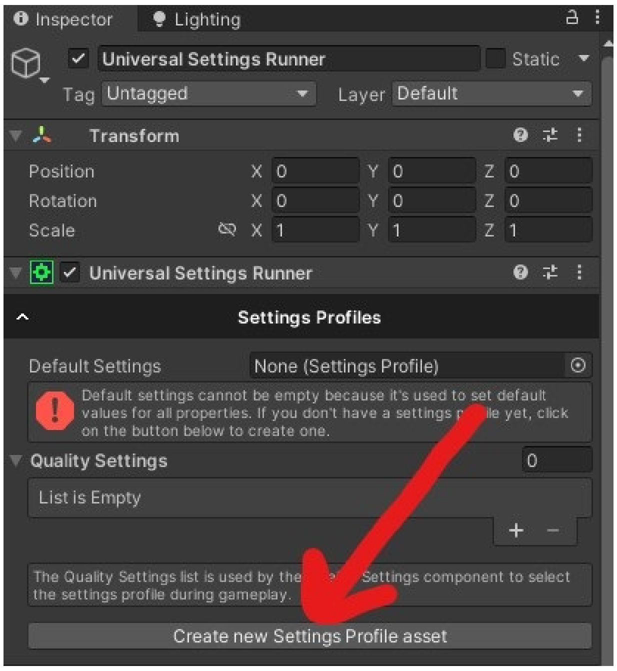
Contents

1 Quick Start

Universal Settings requires only 2 steps.

1. Create a GameObject containing the UniversalSettingsRunner component.
2. Create a Settings Profile. In this file, we will define the initial values for all properties, such as initial brightness intensity, initial fps, etc.




Inspector
Lighting
Universal Settings Runner
Static
Transform
Universal Settings Runner
Settings Profiles
List is Empty
Create new Settings Profile asset

Tag Untagged Layer Default

Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Default Settings None (Settings Profile)

Default settings cannot be empty because it's used to set default values for all properties. If you don't have a settings p[illegible]ile yet, click on the button below to create one.

Quality Settings 0

The Quality Settings list is used by the [illegible] Settings component to select the settings profile during gameplay.

Figure 1: The image on the left represents step 1, while the image on the right represents step 2.

Once you've done that, you have everything you need to build your configuration panel by adding the desired components.

2 Components

All components of the asset, except for the UniversalSettingsRunner, require to be placed alongside another UI component: Toggle, Dropdown, or Slider. At the end of each component's name, it is described which UI component is required for its functioning.

It is possible to define whether the change is automatically applied through the Auto Apply field in all components when manipulating the UI component.

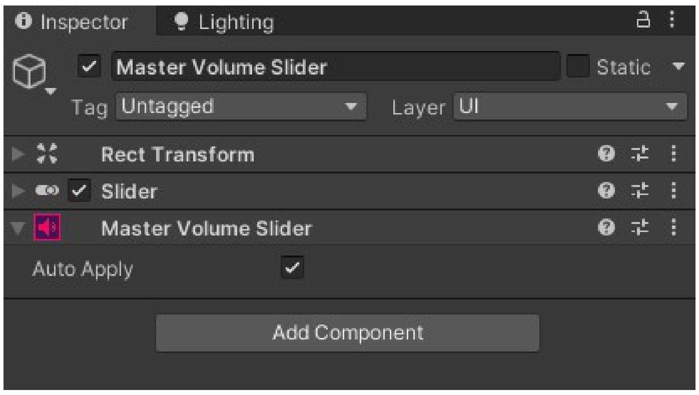
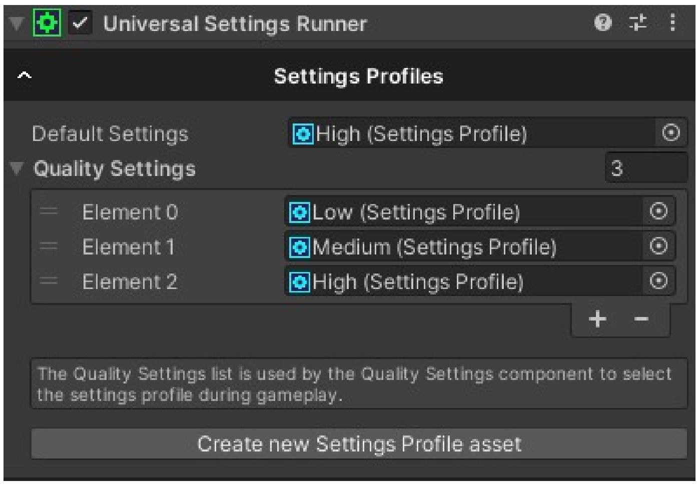


Figure 2: A GameObject containing the Master Volume Slider component.

2.1 Quality Settings Dropdown

This component uses the Quality Settings list defined in the Settings Profile section of the UniversalSettingsRunner component. The component uses all the elements from the list to create the options in the dropdown.

When an item in this dropdown is selected, only the graphics-related properties will be used. These properties are Anti-Aliasing, Shadow Mode, Shadow Distance, Shadow Resolution, Texture Resolution, all Post Processing Effects, and all Renderer Features.



Universal Settings Runner

Settings Profiles

Default Settings: High (Settings Profile)

Quality Settings: 3

Element 0: Low (Settings Profile)

Element 1: Medium (Settings Profile)

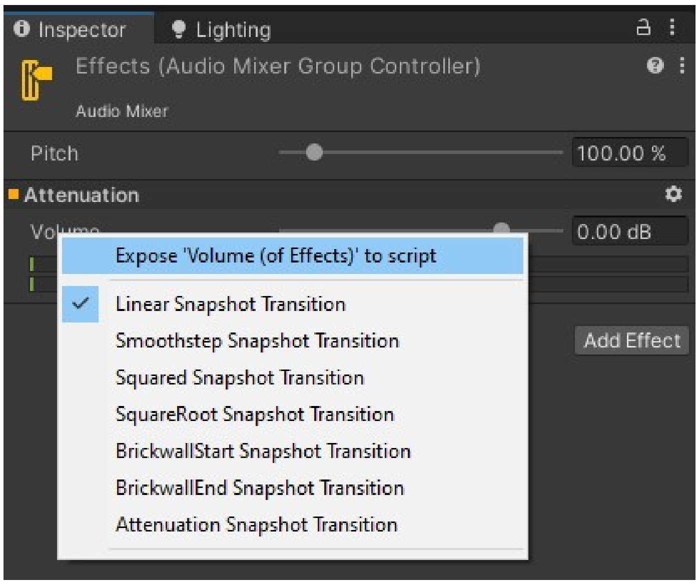
Element 2: High (Settings Profile)

The Quality Settings list is used by the Quality Settings component to select the settings profile during gameplay.

Figure 3: Quality Settings list populated by several Settings Profiles.

2.2 Audio Mixer Volume Slider

To make it work, you need to ensure that the selected Audio Mixer has a group with the volume exposed as a variable. Additionally, you must add the Audio Mixer to the Audio Mixer Configs list in the Audio Mixer section of the UniversalSettingsRunner.


Inspector
Lighting
Effects (Audio Mixer Group Controller)
Audio Mixer
Attenuation
Expose 'Volume (of Effects)' to script
Linear Snapshot Transition
Smoothstep Snapshot Transition
Squared Snapshot Transition
SquareRoot Snapshot Transition
BrickwallStart Snapshot Transition
BrickwallEnd Snapshot Transition
Attenuation Snapshot Transition
Add Effect

Pitch 100.00 %

Volume 0.00 dB

Figure 4: Exposing the variable volume of an Audio Mixer Group Controller.

2.3 Brightness Slider

This component uses post-processing to control brightness. If you are using the Built-in pipeline, you need to add the Color Grading effect to the post-processing profile, or Color Adjustments in the case of the URP pipeline.

Remember that the post-processing profile must be attached to the UniversalSettingsRunner in the Post Processing section.

2.4 Post Processing Effect Toggle

Its function is to enable or disable a post-processing effect. The selected effect in this component must be present in one or more post-processing profiles defined in the UniversalSettingsRunner component.

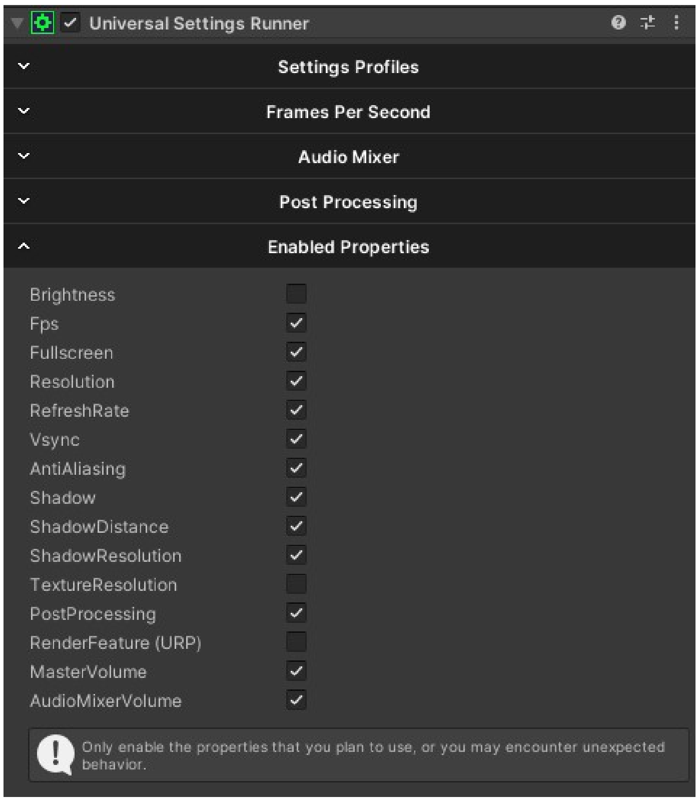
2.5 Others components

No additional configuration is required for the remaining components that have not been mentioned. Simply add them to a GameObject, and they will work seamlessly.

3 Enabled Properties

All properties are enabled by default, but it is possible to disable them if they are not planned to use. For example, if your game does not have brightness control, you can disable it in the Enabled Properties section of the UniversalSettingsRunner.

A disabled property is completely ignored by Universal Settings, saving processing power and allowing you to manipulate it manually if desired.



Brightness	
Fps	✓
Fullscreen	✓
Resolution	✓
RefreshRate	✓
Vsync	✓
AntiAliasing	✓
Shadow	✓
ShadowDistance	✓
ShadowResolution	✓
TextureResolution	
PostProcessing	✓
RenderFeature (URP)	
MasterVolume	✓
AudioMixerVolume	✓

Only enable the properties that you plan to use, or you may encounter unexpected behavior.

Figure 5: Enabled Properties section with some properties disabled.

4 Callback and Custom Components

There is a callback called onApplySettings in the UniversalSettingsRunner class that is triggered whenever there is a change in the settings.

The logic for custom components such as Custom Boolean Toggle, Custom Float Slider, and Custom Integer Dropdown can be implemented using the callback, as shown in the following code snippet:

```
UniversalSettingsRunner.Instance.onApplySettings += () => {
    bool somethingBoolean = UniversalSettingsRunner.Instance.GetCustomBoolean(0);
    float somethingFloat = UniversalSettingsRunner.Instance.GetCustomFloat(0);
    int somethingInt = UniversalSettingsRunner.Instance.GetCustomInteger(0);

    // You can do whatever you want with the custom settings here.
};
```