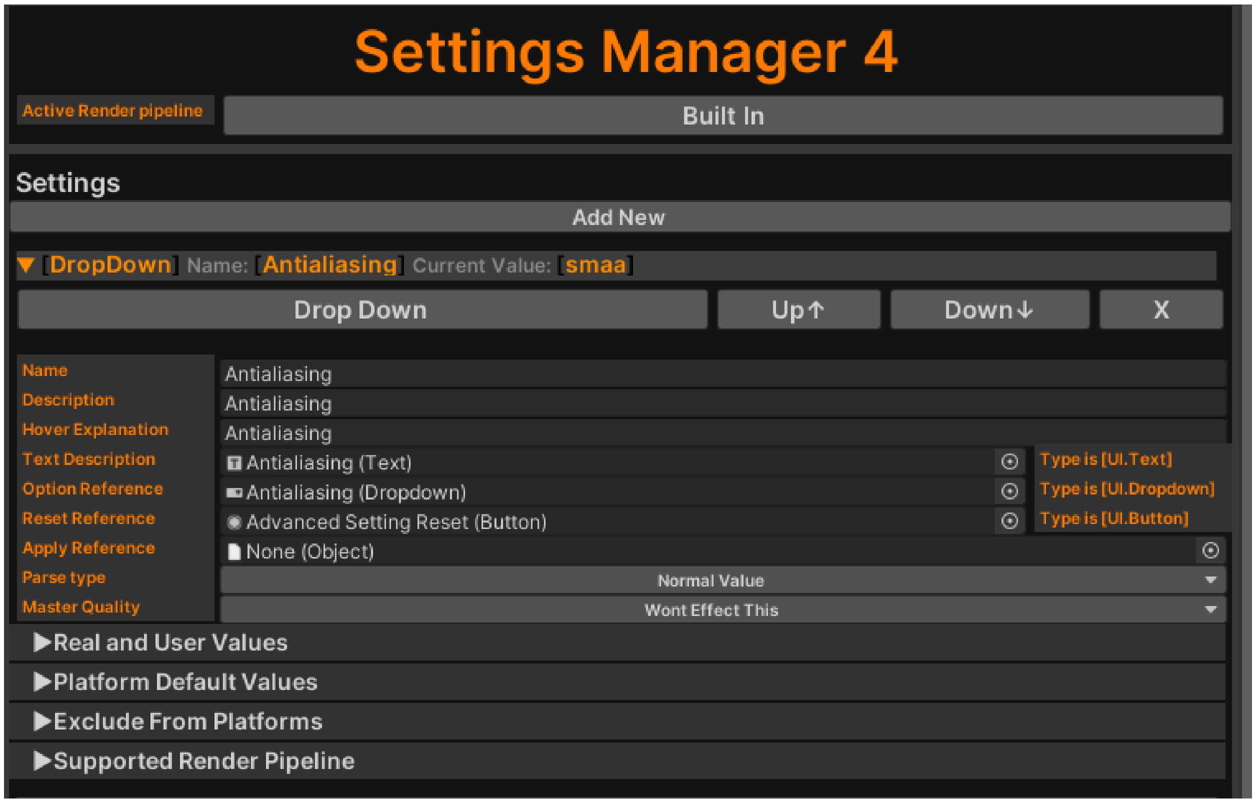
Settings Manager Documentation

Version 4



Active Render pipeline: Built In

▼ [DropDown] Name: [Antialiasing] Current Value: [smaa]

Name	Antialiasing	
Description	Antialiasing	
Hover Explanation	Antialiasing	
Text Description	Antialiasing (Text)	Type is [UI.Text]
Option Reference	Antialiasing (Dropdown)	Type is [UI.Dropdown]
Reset Reference	Advanced Setting Reset (Button)	Type is [UI.Button]
Apply Reference	None (Object)	
Parse type	Normal Value	
Master Quality	Wont Effect This	

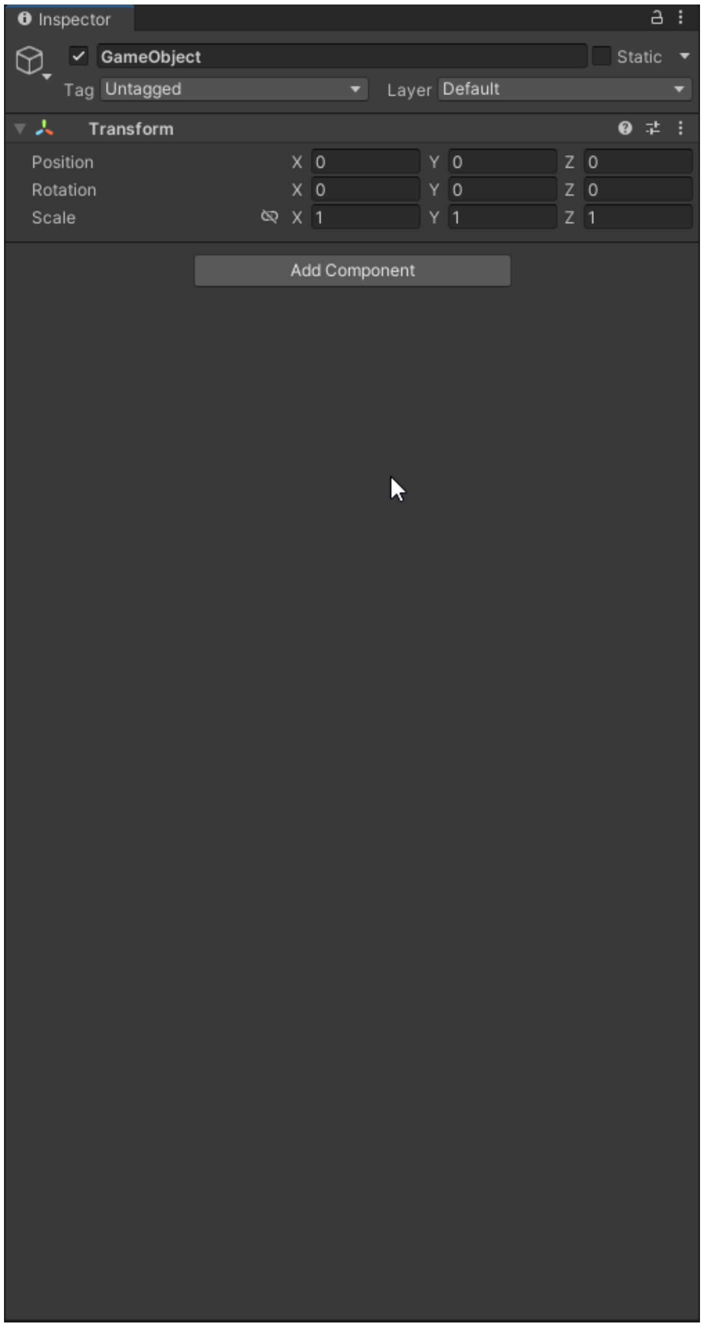
A settings manager designed for quick and easy implementation of user-facing options

Please ensure you're viewing this documentation on the google doc link for the up-to-date Giffify version. Settings Manager Documentation 4

Installation

Quickstart

To install Settings Manager, simply add the Manager script to a game object in your scene. All settings Manager-related scripts can be found under **Add Component > BattlePhaze > Settings Manager.** Note there are also some optional scripts that won't be added automatically, these include Modules, Input Types, and Manager input types.



Unity version support

The settings manager currently supports unity 2019,2021,2022 and above. Some features are only available on certain platforms and unity versions.

Dependencies

Settings manager 4 and above do not require any other packages, however, TextMeshPro is recommended.

Demo scene

Settings Manager comes with a Demo scene located at **Assets>BattlePhazeSettingsManager> SettingsManagerDemo**. This scene demonstrates a proper setup for the settings manager and shows how all the parts work together. We will be using this scene as an example throughout the documentation.

Overview

Version Number

The current version of Settings Manager.

Render Pipeline

Currently selected render pipeline. Settings Manager supports the High Definition Render Pipeline, Built-In, and Universal Render Pipeline.

Settings

Listed here are all the settings that the settings manager is handling for the game.

Manager Settings

Settings for Settings Manager itself and how it operates.

Module Settings

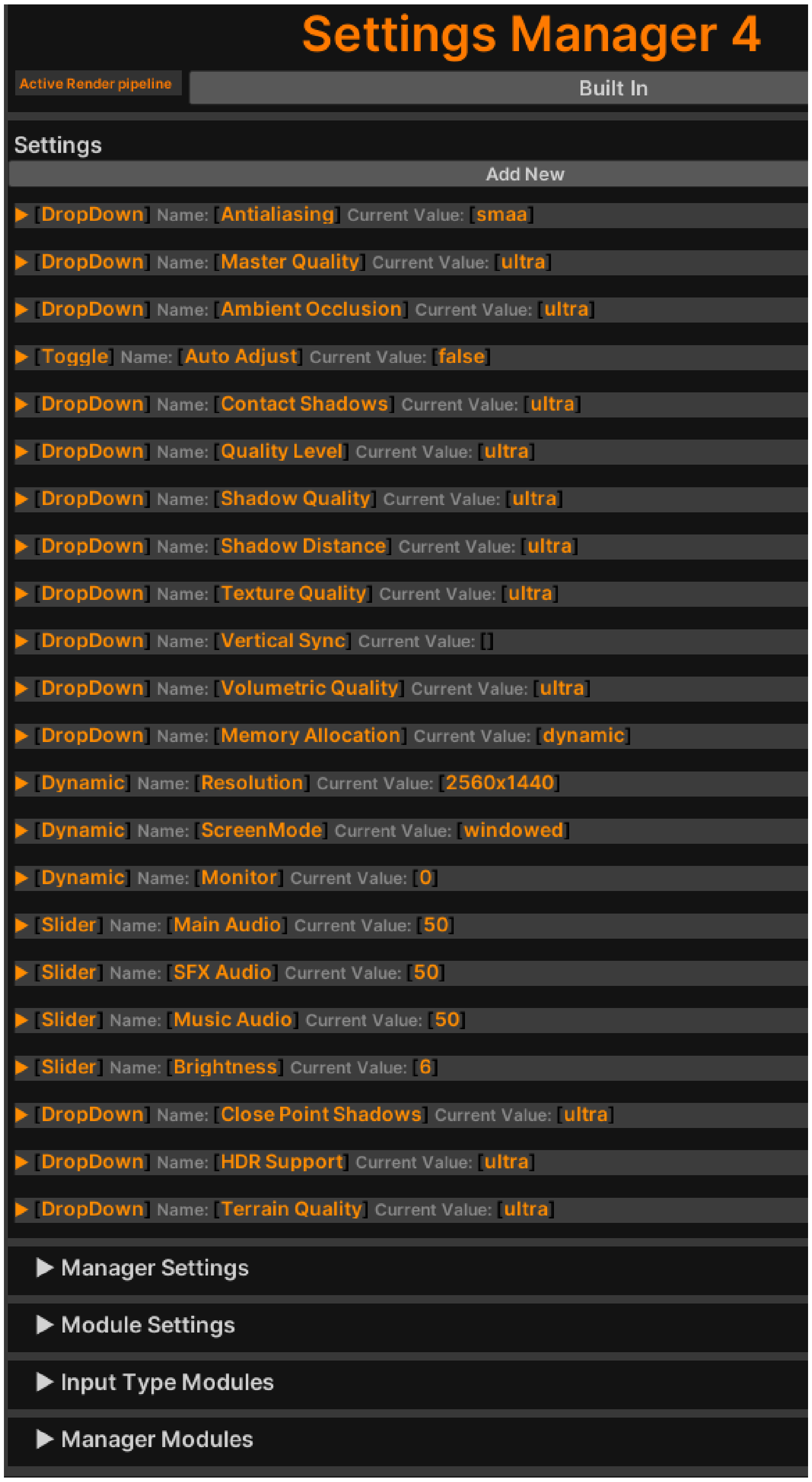
Configuration for modules installed into settings manager for managing different in-game settings.

Input type Modules

Configuration for input types within the settings manager, such as Unity UI, TextMeshPro, and Michsky.

Manager Modules

Manager Modules handle assigning listeners for specific functions to specific buttons in Settings Manager. This is used in tandem with the Input type modules so that the settings manager can assign the correct listeners for the different input types.



Settings Manager 4

Active Render pipeline: Built In

Settings

Add New

▶ [DropDown] Name: [Antialiasing] Current Value: [smaa]
▶ [DropDown] Name: [Master Quality] Current Value: [ultra]
▶ [DropDown] Name: [Ambient Occlusion] Current Value: [ultra]
▶ [Toggle] Name: [Auto Adjust] Current Value: [false]
▶ [DropDown] Name: [Contact Shadows] Current Value: [ultra]
▶ [DropDown] Name: [Quality Level] Current Value: [ultra]
▶ [DropDown] Name: [Shadow Quality] Current Value: [ultra]
▶ [DropDown] Name: [Shadow Distance] Current Value: [ultra]
▶ [DropDown] Name: [Texture Quality] Current Value: [ultra]
▶ [DropDown] Name: [Vertical Sync] Current Value: []
▶ [DropDown] Name: [Volumetric Quality] Current Value: [ultra]
▶ [DropDown] Name: [Memory Allocation] Current Value: [dynamic]
▶ [Dynamic] Name: [Resolution] Current Value: [2560x1440]
▶ [Dynamic] Name: [ScreenMode] Current Value: [windowed]
▶ [Dynamic] Name: [Monitor] Current Value: [0]
▶ [Slider] Name: [Main Audio] Current Value: [50]
▶ [Slider] Name: [SFX Audio] Current Value: [50]
▶ [Slider] Name: [Music Audio] Current Value: [50]
▶ [Slider] Name: [Brightness] Current Value: [6]
▶ [DropDown] Name: [Close Point Shadows] Current Value: [ultra]
▶ [DropDown] Name: [HDR Support] Current Value: [ultra]
▶ [DropDown] Name: [Terrain Quality] Current Value: [ultra]

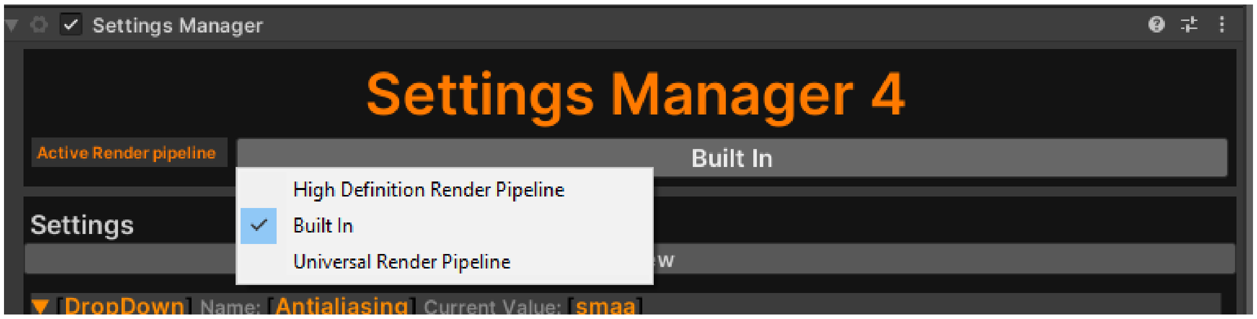
▶ Manager Settings

▶ Module Settings

▶ Input Type Modules

▶ Manager Modules

Render Pipeline

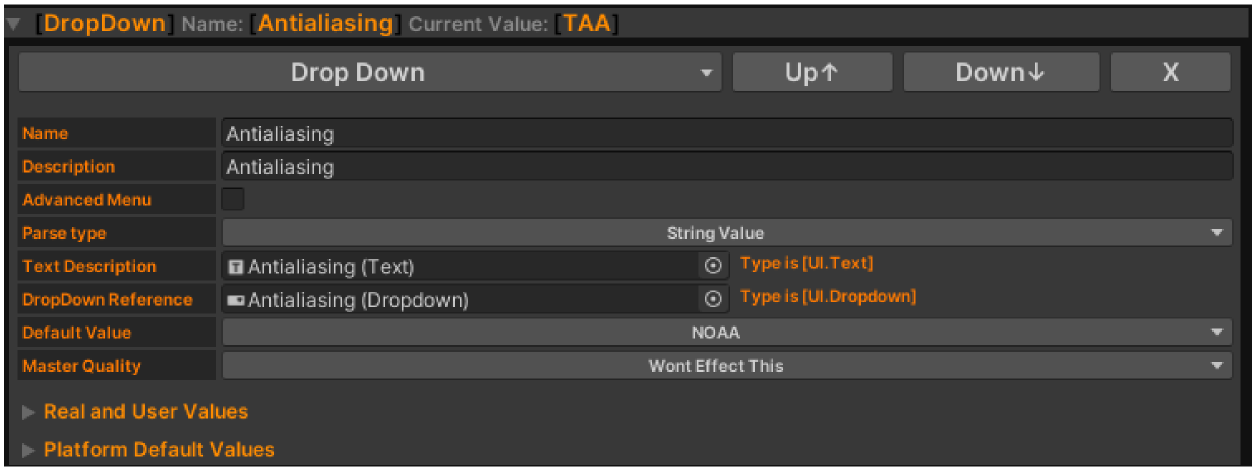

Settings Manager
Settings Manager 4
Active Render pipeline
Built In
High Definition Render Pipeline
Built In
Universal Render Pipeline
Settings
[DropDown] Name: [Antialiasing] Current Value: [smaa]

The selected pipeline can be changed by changing the drop-down at the top of Settings Manager. Currently, the Settings Manager Supports
High Definition Render Pipeline, Built-In, Universal Render Pipeline.

Custom versions of SRP are also supported!

make sure to swap to the correct pipeline you are on. this allows scripts to compile for the correct pipeline.

Anatomy of a Setting



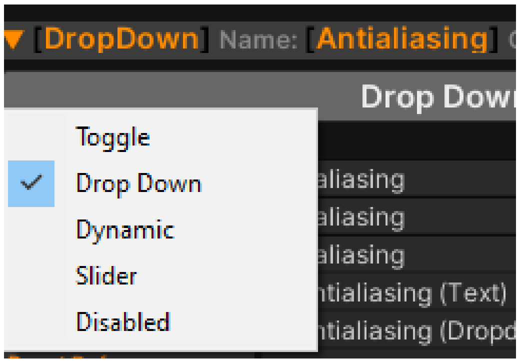
[DropDown] Name: [Antialiasing] Current Value: [TAA]

Name	Antialiasing
Description	Antialiasing
Advanced Menu	
Parse type	String Value
Text Description	Antialiasing (Text) — Type is [UI.Text]
DropDown Reference	Antialiasing (Dropdown) — Type is [UI.Dropdown]
Default Value	NOAA
Master Quality	Wont Effect This

Real and User Values

Platform Default Values

Input type



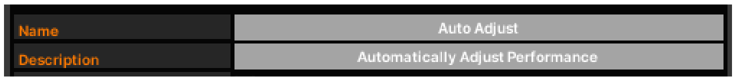
The first option you will see when opening a setting is the type of input the setting accepts. The options listed here depend on the input type modules installed. In the demo scene, you can see we have the options of Toggle, Drop Down, Dynamic, Slider, and Disabled. The settings Type will also be displayed at the top of the setting next to its name and current value.

Sorting arrows, Up down and X

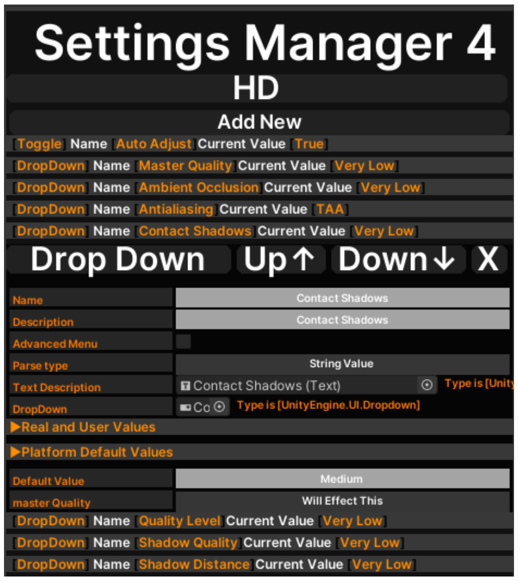
The **Up** and **Down** arrows allow you to organize and group settings together within the editor window. The **X** button will remove that setting. Note the order of the settings within the Settings Manager editor does not affect the order of the settings displayed in your game.

Name

This is how we identify a setting on a module.



Name	Auto Adjust
Description	Automatically Adjust Performance



Settings Manager 4

HD

Add New

[Toggle] Name [Auto Adjust] Current Value [True]
[DropDown] Name [Master Quality] Current Value [Very Low]
[DropDown] Name [Ambient Occlusion] Current Value [Very Low]
[DropDown] Name [Antialiasing] Current Value [TAA]
[DropDown] Name [Contact Shadows] Current Value [Very Low]

Name	Contact Shadows
Description	Contact Shadows
Advanced Menu	
Parse type	String Value
Text Description	Contact Shadows (Text) — Type is [Unity
DropDown	Co — Type is [UnityEngine.UI.Dropdown]

▶Real and User Values

▶Platform Default Values

Default Value	Medium
master Quality	Will Effect This

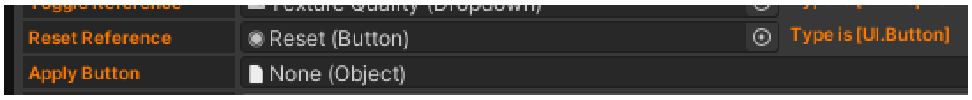
[DropDown] Name [Quality Level] Current Value [Very Low]
[DropDown] Name [Shadow Quality] Current Value [Very Low]
[DropDown] Name [Shadow Distance] Current Value [Very Low]

Description

The text will be displayed beside the option.

Reset and Apply

The Apply button updates all options & saves the options change to file.
The Reset reference (normally a button) will reset the option back to the default.



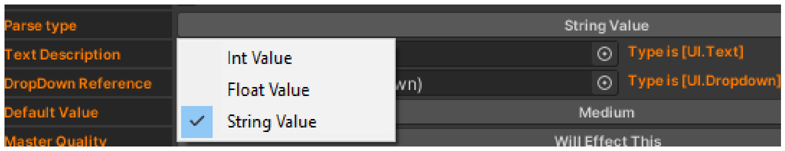
Reset Reference	Reset (Button)	Type is [UI.Button]
Apply Button	None (Object)	

Hover Explanation

Text entered here will appear when a user hovers over a setting in the game. **You must assign a Text Field to the Hover Explanation option in the Manager settings for this to work.**

Parse Type

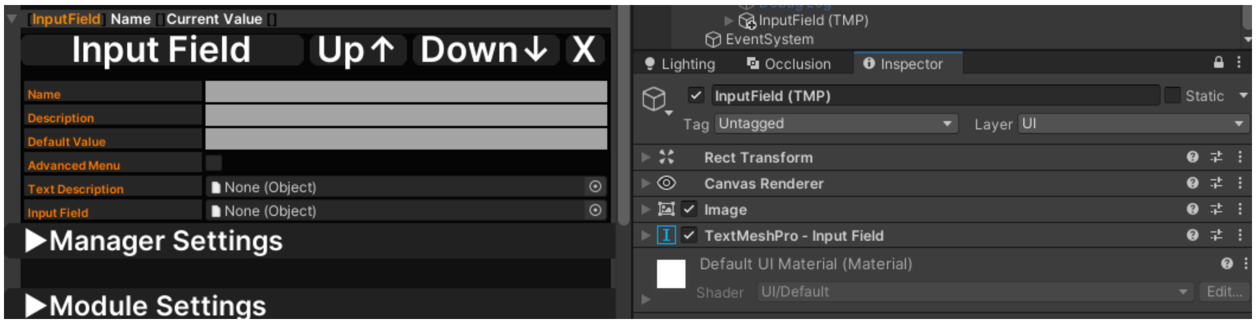
The variable type is parsed for the option in question. Available options are Normal Value, and Int.



Parse type: String Value
Text Description: Type is [UI.Text]
DropDown Reference: wn) Type is [UI.Dropdown]
Default Value: Medium
Master Quality: Will Effect This

UI Object Fields

Depending on the type of setting you are creating (Toggle, Slider, etc.) you will have different options here for assigning a Unity UI object. **Dragging a game object over to these fields will attempt to find the reference to the UI object but it might give false positives.** These are the UI objects that appear in the game and will control that setting.



InputField Name Current Value
Input Field Up↑ Down↓ X

Name:
Description:
Default Value:
Advanced Menu:
Text Description: None (Object)
Input Field: None (Object)

▶Manager Settings
▶Module Settings

InputField (TMP)
EventSystem

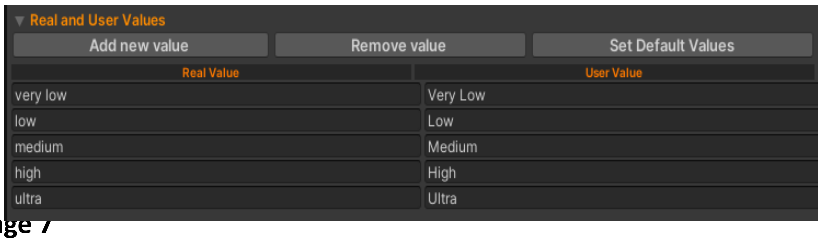
Lighting | Occlusion | Inspector

✓ InputField (TMP) Static
Tag Untagged Layer UI

Rect Transform
Canvas Renderer
✓ Image
✓ TextMeshPro - Input Field
Default UI Material (Material)
Shader UI/Default Edit...

Real and User Values

If the setting you are creating is a Drop Down or Dynamic, you will have the option for Real and User values. Real values



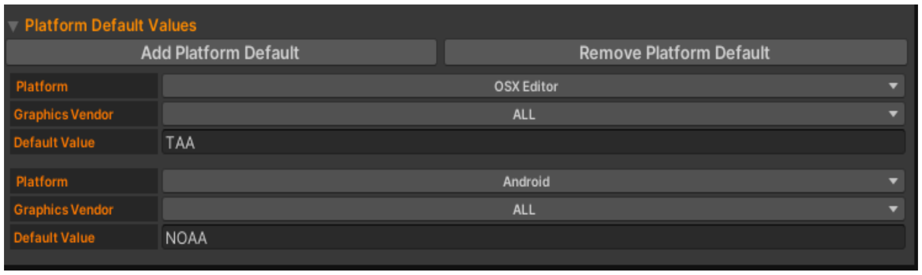
Real and User Values

Real Value	User Value
very low	Very Low
low	Low
medium	Medium
high	High
ultra	Ultra

are code-facing while User values are the values displayed to the end user. (for example, we save the Real Value to file but Show the User Value in the UI)

Platform Default Values

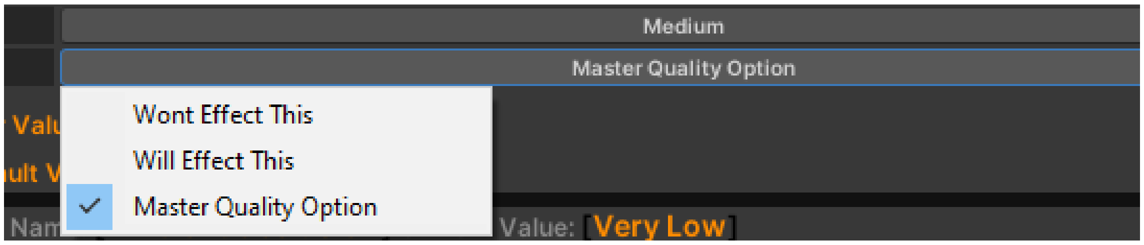
You can set the Default Value per platform and graphics vendor if the setting you're creating is a Toggle, Drop Down, or Slider.



Platform	OSX Editor
Graphics Vendor	ALL
Default Value	TAA
Platform	Android
Graphics Vendor	ALL
Default Value	NOAA

Default Value

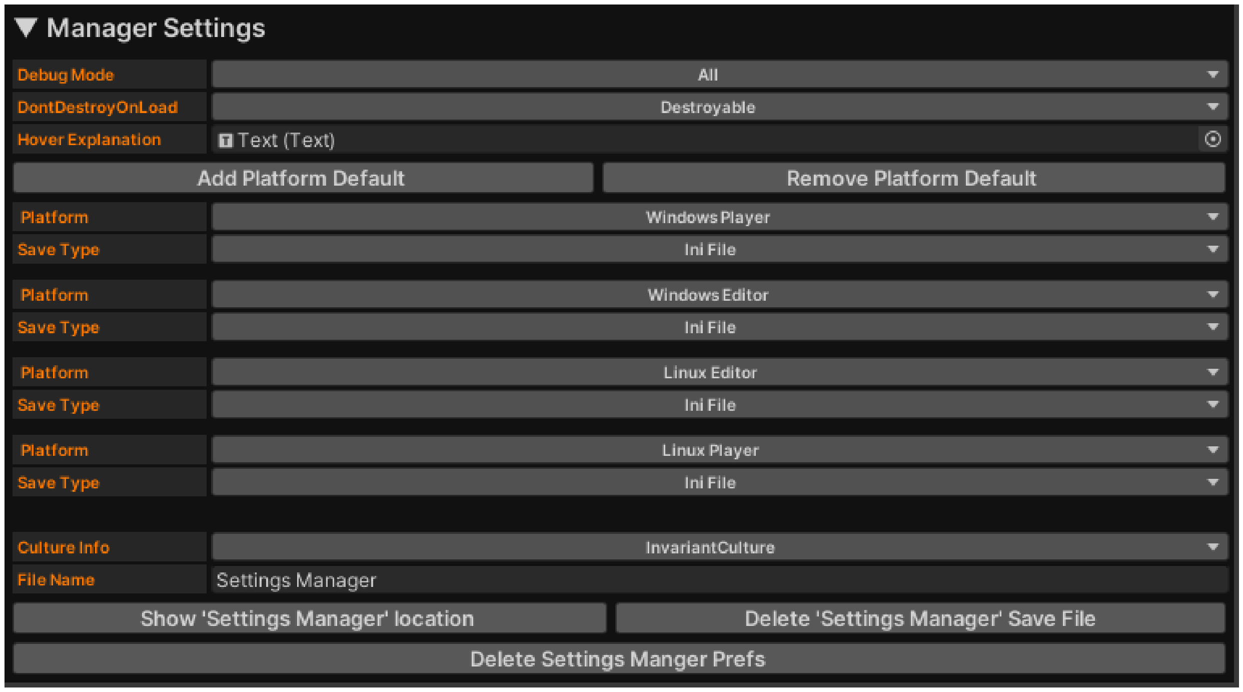
For dropdowns, if there are no Platform Default Values set for a given platform then the Default value will be used instead.



Master Quality

Settings Manager allows you to create a Master Quality Setting to change the options on many settings at once. To create the Master Quality Setting, create a new setting and then select 'Master Quality Option' under the master quality dropdown. This setting will then control all other settings that have the master quality dropdown set to Enabled.

Manager Settings



Debug Mode	All
DontDestroyOnLoad	Destroyable
Hover Explanation	Text (Text)

Add Platform Default
Remove Platform Default

Platform	Windows Player
Save Type	Ini File
Platform	Windows Editor
Save Type	Ini File
Platform	Linux Editor
Save Type	Ini File
Platform	Linux Player
Save Type	Ini File
Culture Info	InvariantCulture
File Name	Settings Manager

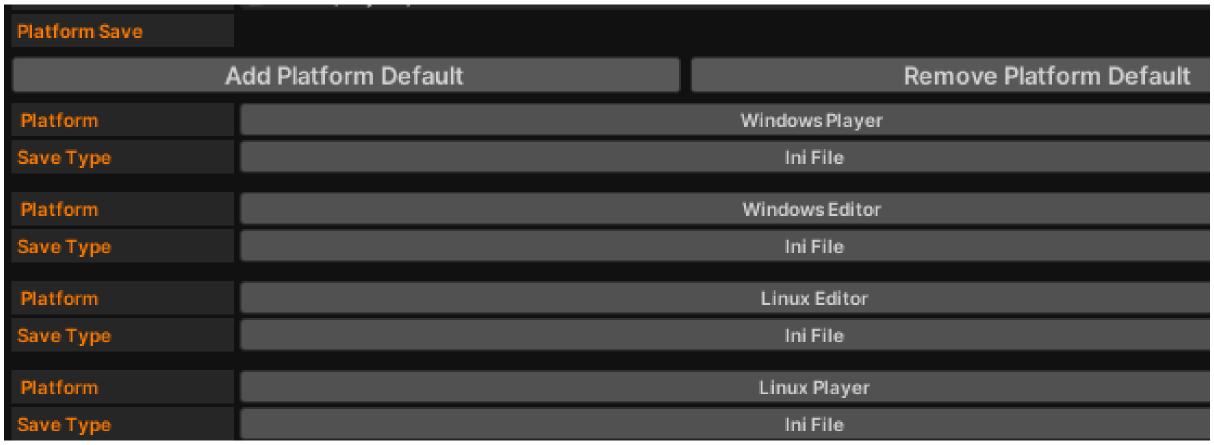
Show 'Settings Manager' location
Delete 'Settings Manager' Save File
Delete Settings Manger Prefs

Hover Explanation

Drag a Text field here to have it appear when a user hovers over a setting in the game. The text displayed is configured in the options for each setting individually.

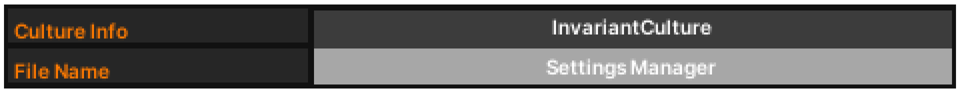
Saving

To preserve the user-selected settings between sessions, the settings are saved to storage. this is done through a save module.
You can select here which save module per platform you would like to use.



Platform Save	
Add Platform Default	Remove Platform Default
Platform	Windows Player
Save Type	Ini File
Platform	Windows Editor
Save Type	Ini File
Platform	Linux Editor
Save Type	Ini File
Platform	Linux Player
Save Type	Ini File

Culture Info



Culture Info	InvariantCulture
File Name	Settings Manager

The culture settings adjust how the data for the settings manager is saved and read to account for differences in formatting such as using a period "." versus a comma "," to indicate a decimal point. note that **Invariant Culture** will be the one to use most of the time.

Invariant Culture

the invariant culture is culture insensitive (it is associated with the English language but not with any country/region

Current Culture

CurrentCulture is the .NET representation of the default user locale of the system. This controls default number and date formatting and the like.

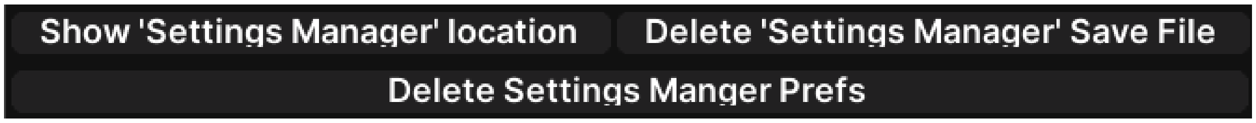
Current UI Culture

CurrentUICulture refers to the default user interface language, a setting introduced in Windows 2000. This is primarily regarding the UI localization/translation part of your app.

Installed UI Culture

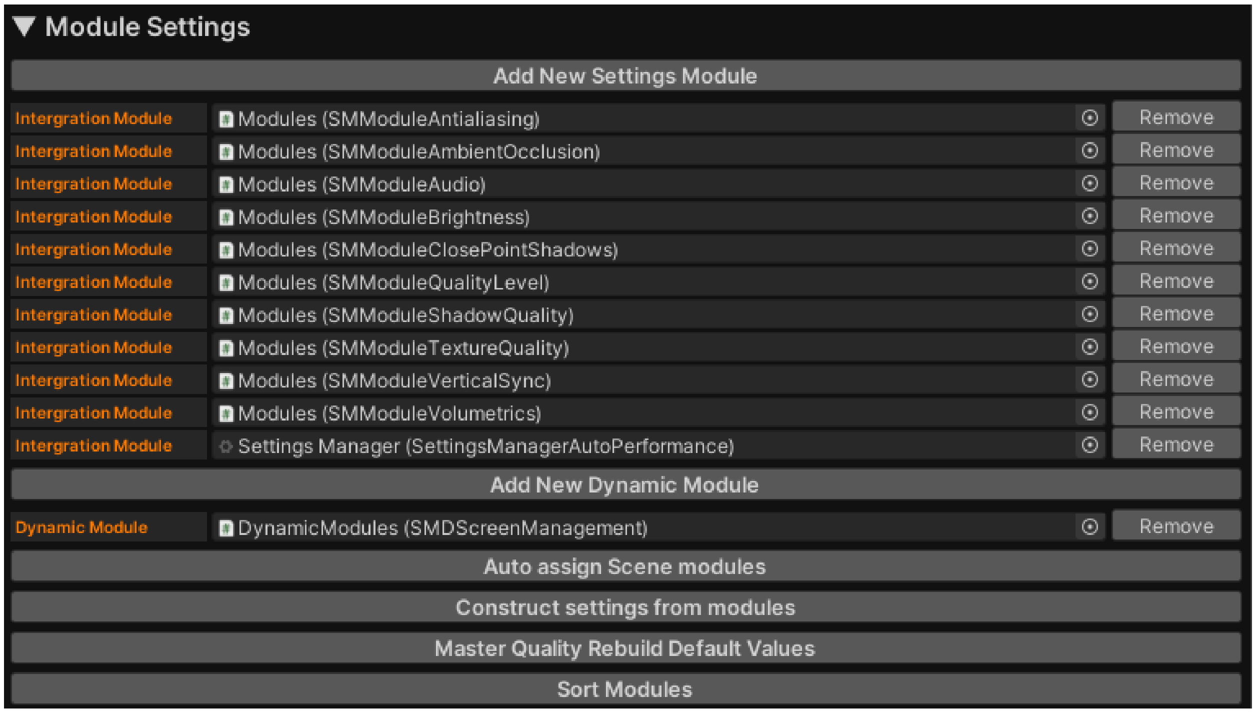
Gets the CultureInfo that represents the culture installed with the operating system.

Data Management


Show 'Settings Manager' location
Delete 'Settings Manager' Save File
Delete Settings Manger Prefs

Here you have options to open the folder that contains the Ini that stores all the settings, an option to delete that file, and an option to delete the settings stored in the PlayerPrefs. As these options are only available in the unity editor, this is mainly intended for debugging.

Module Settings


▼ Module Settings

Add New Settings Module

Intergration Module	Modules (SMModuleAntialiasing)	Remove
Intergration Module	Modules (SMModuleAmbientOcclusion)	Remove
Intergration Module	Modules (SMModuleAudio)	Remove
Intergration Module	Modules (SMModuleBrightness)	Remove
Intergration Module	Modules (SMModuleClosePointShadows)	Remove
Intergration Module	Modules (SMModuleQualityLevel)	Remove
Intergration Module	Modules (SMModuleShadowQuality)	Remove
Intergration Module	Modules (SMModuleTextureQuality)	Remove
Intergration Module	Modules (SMModuleVerticalSync)	Remove
Intergration Module	Modules (SMModuleVolumetrics)	Remove
Intergration Module	Settings Manager (SettingsManagerAutoPerformance)	Remove

Add New Dynamic Module

Dynamic Module	DynamicModules (SMDScreenManagement)	Remove

Auto assign Scene modules

Construct settings from modules

Master Quality Rebuild Default Values

Sort Modules

Modules

The following is a revised version of the original text:

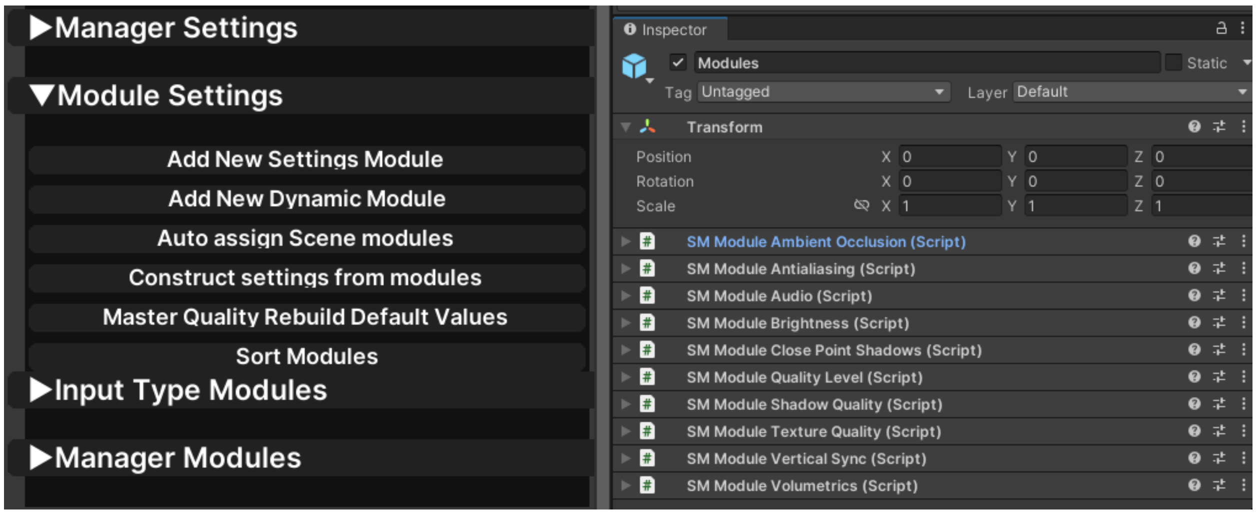
The Settings Manager modules are responsible for adjusting the application based on user inputs from the settings menu. There are two types of modules: Static and Dynamic. Static Modules have predetermined options, such as Antialiasing, which only needs a set number of options to choose from. Dynamic Modules construct their options at runtime. For example, screen resolutions must be determined per device unless every possible resolution is manually coded.

Each module is defined as a class that inherits from SettingsManagerOption.

Add New Settings Module

To integrate a module into the Settings Manager, start by adding the module script to a game object in the scene. Then, click on the "Add New Settings Module" button and drag the script for the module into the designated field. An alternative option is to use the "Auto-assign Scene Modules" button, which searches the scene hierarchy and automatically assigns any modules it finds.

The premade module scripts are located in a folder called SettingsManagerModules



Modules ✓ Static

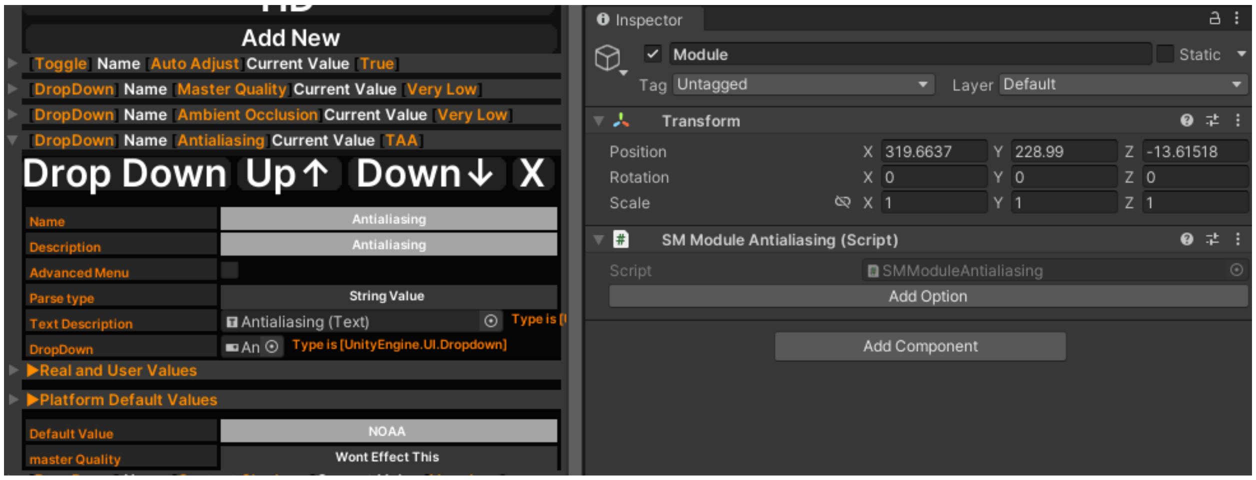
Tag Untagged Layer Default

Transform

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

SM Module Ambient Occlusion (Script)
SM Module Antialiasing (Script)
SM Module Audio (Script)
SM Module Brightness (Script)
SM Module Close Point Shadows (Script)
SM Module Quality Level (Script)
SM Module Shadow Quality (Script)
SM Module Texture Quality (Script)
SM Module Vertical Sync (Script)
SM Module Volumetrics (Script)

After adding a module you must assign it to an option in the settings manager so that when a user changes an option settings manager knows which module it corresponds to.


Add New

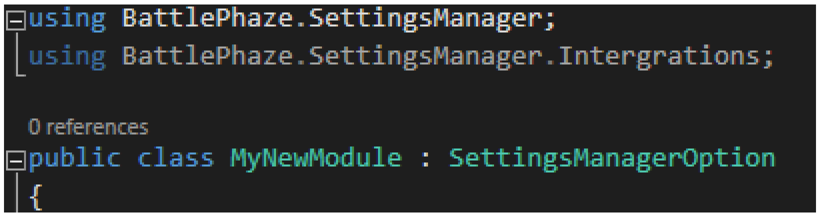
Toggle Name Auto Adjust Current Value True
DropDown Name Master Quality Current Value Very Low
DropDown Name Ambient Occlusion Current Value Very Low
DropDown Name Antialiasing Current Value TAA

Drop Down Up↑ Down↓ X

Name: Antialiasing
Description: Antialiasing
Advanced Menu
Parse type: String Value
Text Description: Antialiasing (Text) Type is [
DropDown: An Type is [UnityEngine.UI.Dropdown]
▶Real and User Values
▶Platform Default Values
Default Value: NOAA
master Quality: Wont Effect This

Inspector
Module Static
Tag Untagged Layer Default
Transform
Position X 319.6637 Y 228.99 Z -13.61518
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
SM Module Antialiasing (Script)
Script SMModuleAntialiasing
Add Option
Add Component

Creating A Module

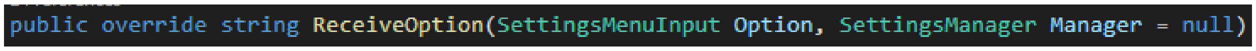

```
using BattlePhaze.SettingsManager;
using BattlePhaze.SettingsManager.Intergrations;

0 references
public class MyNewModule : SettingsManagerOption
{
```

Settings Manager comes with many modules for common settings, however, you can also create your own modules for adjusting any other options you want in your application.

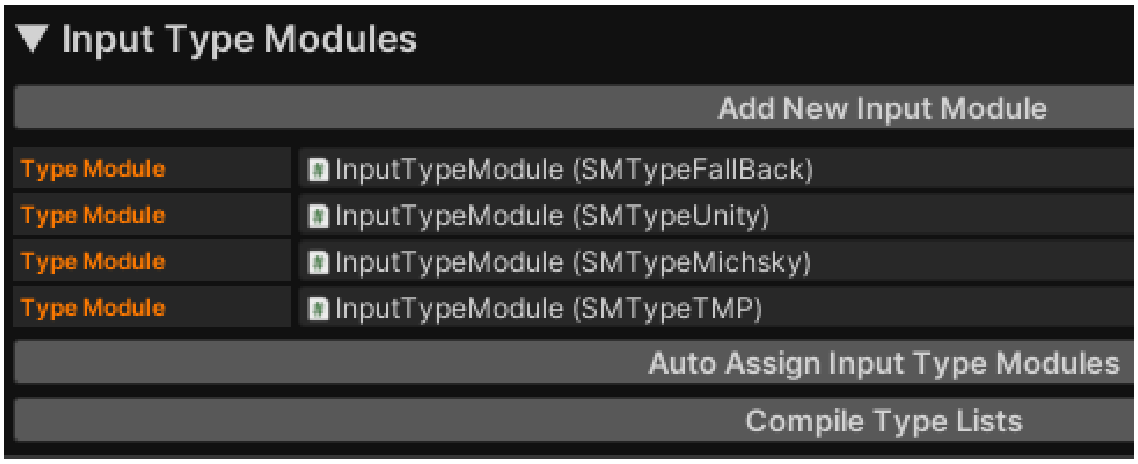
To create a Static module, your class must inherit from SettingsManagerOption. To create a Dynamic module, your class will inherit from SettingsManagerDynamicOption instead. Once you have your class, you need to implement some overrides for it to work as a module for the settings manager. Static modules require the following overrides:

ReceiveOption


```
public override string ReceiveOption(SettingsMenuInput Option, SettingsManager Manager = null)
```

ReceiveOption is called when a user selects an option in the in-game settings. This is where you will determine the value of the selected option and adjust the game accordingly. Take a look at the SMModuleShadowQuality.cs module for an example.

Input Type Modules

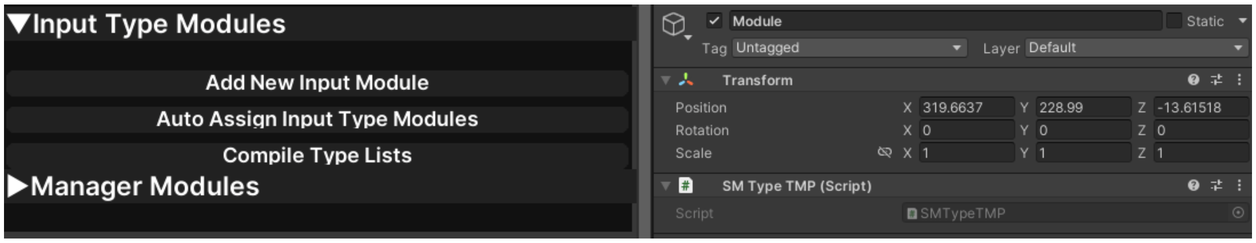


Type Module	InputTypeModule (SMTypeFallBack)
Type Module	InputTypeModule (SMTypeUnity)
Type Module	InputTypeModule (SMTypeMichsky)
Type Module	InputTypeModule (SMTypeTMP)

Input-type modules handle receiving input from various UI elements. Settings Manager comes with modules to support Unity's standard UI, Textmesh Pro, and Michsky. Depending on the UI system you are using for your settings menu you will need its corresponding module assigned in the input type modules section of Settings Manager.

Adding Input Types

To add the input type for your UI solution add the corresponding script for the module to a game object in the scene. Input type modules are located at Assets > BattlePhazeSettingsManager > SettingsManager > SettingsmanagerTypeModules > SMType. Then assign it to Settings Manager by dragging the component into the input type modules section.

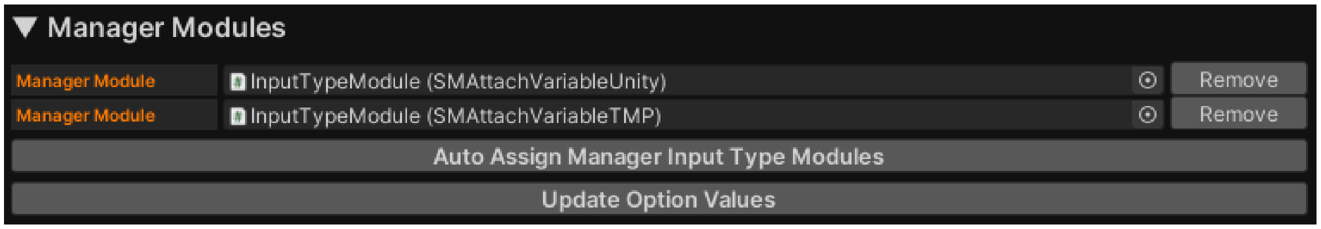


Tag: Untagged Layer: Default
Position X 319.6637 Y 228.99 Z -13.61518
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Script: SMTypeTMP

Compile Type Lists

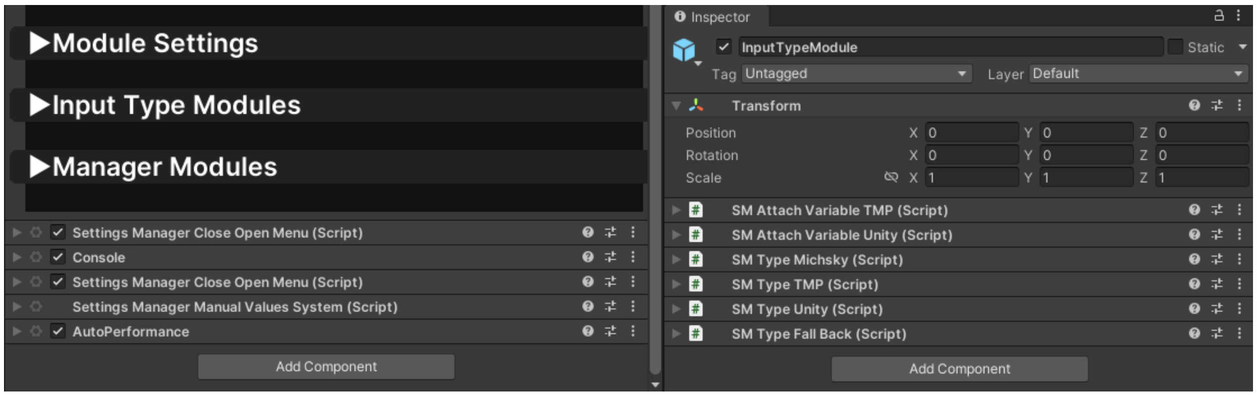
This button will go through all of the input type modules and add their corresponding Scripting Define Symbols to the Unity Player Settings.

Manager Modules

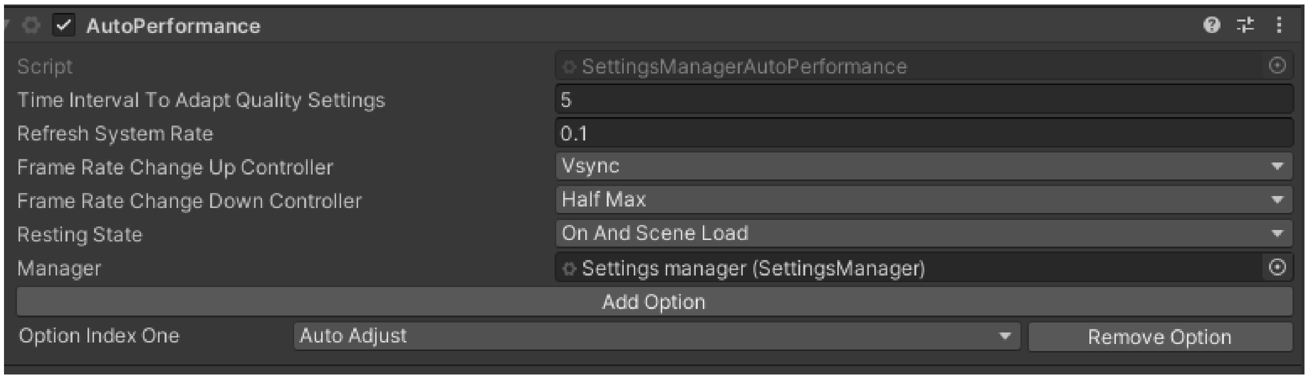


Manager Module: InputTypeModule (SMAttachVariableUnity)
Manager Module: InputTypeModule (SMAttachVariableTMP)

Manager Modules handle assigning listeners for specific functions to specific buttons in Settings Manager. Generally, you will never need to do any more than assign the SM Attach Variable Modules unless you are making an addon for the settings manager. In the demo scene, you can see we have all of the Input Type Modules alongside the SM Attach Variable Modules on one game object, and then just auto-assign them with the Auto-assign buttons in the corresponding sections of Settings Manager.



Auto Performance System


AutoPerformance

Script	SettingsManagerAutoPerformance
Time Interval To Adapt Quality Settings	5
Refresh System Rate	0.1
Frame Rate Change Up Controller	Vsync
Frame Rate Change Down Controller	Half Max
Resting State	On And Scene Load
Manager	Settings manager (SettingsManager)

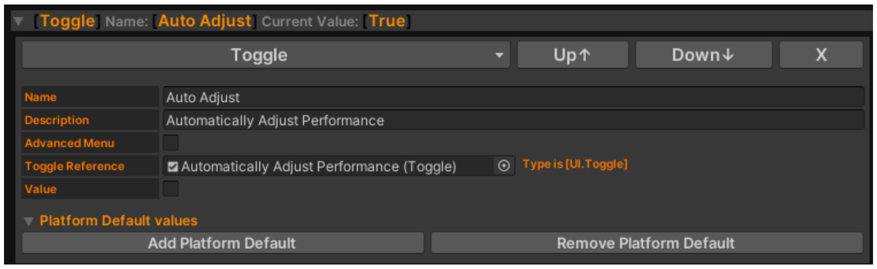
Option Index One: Auto Adjust

The Auto Performance component allows Settings Manager to automatically adjust settings to reach a specified performance target. The Auto Performance system is built on top of the

Module system so it acts just like any other Module and will need to be assigned under the Module Settings.

Setup

To set this up you will need an option in your in-game settings menu and a corresponding setting in the settings manager. You will also need to have the Master Quality option set up as the Auto Adjust component will apply master quality levels. Take a Look at the Demo Scene and the Master Quality explanation in the Anatomy of a Setting section of this documentation.



[Toggle] Name: [Auto Adjust] Current Value: [True]

Name	Auto Adjust
Description	Automatically Adjust Performance
Advanced Menu	☐
Toggle Reference	☑ Automatically Adjust Performance (Toggle) Type is [UI.Toggle]
Value	☐

Platform Default values

Options

Name: This is the name of the setting in Settings Manager that controls the Auto Performance system. In the demo scene, we have this setup as a toggle to turn auto performance on and off.
Explanation: This fills in the Description section of the corresponding Setting if you use the Construct settings from modules button in the Module Settings section.
System Startup State: The default state for the option. If set to on then the Auto Performance setting will be on when the application starts. In most cases, you would want this off.
Up Threshold: The Point at which the Master quality settings will be adjusted up one level, improving visual quality at the cost of performance.
Down threshold: The point at which the Master Quality settings will be adjusted, reducing visual quality to improve performance.

Contact Info

If you have any issues, or questions, or are looking for further support please reach out to me.

Luke Dooly
Email | doolanl208@gmail.com
Discord | dooly#0001 | https://discord.gg/bEzmAbC