

“No Frills” Modifiers

“No Frills” Modifiers

namespace GWLPXL.NoFrills.Modifiers.com

A simple “No Frills” system designed to help you add and manage **Modifiers**!

Modifiers are ways to change how your game plays! These **Modifiers** are number-based and modify numbers at runtime. A common example is a stat system, you have a base Strength stat of 5, but then you wear equipment that provide +5 more strength, i.e. a modifier.

Table of Contents

[Package Details](#)

[BaseModHolder](#)

[Modifiable](#)

[Create Generic Mod Holder](#)

[Create Modifiable](#)

[Extending the Package](#)

[Create New from ModBase](#)

[Create New From BaseModHolder](#)

[F.A.Q.](#)

Other Links

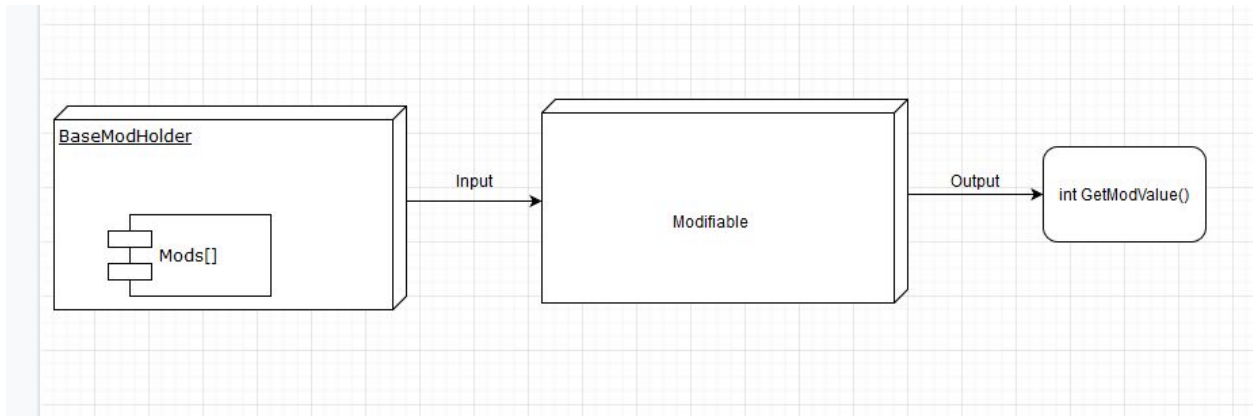
- ❖ [Online Documentation Link \(most up to date info\)](#)
- ❖ [Online Feedback Form](#)

“No Frills” Modifiers

Package Details

The core components of the **Modifiers** package.

Pass in a **BaseModHolder** to add or remove them from the **Modifiable**. The Modifiable outputs the value.



“No Frills” Modifiers

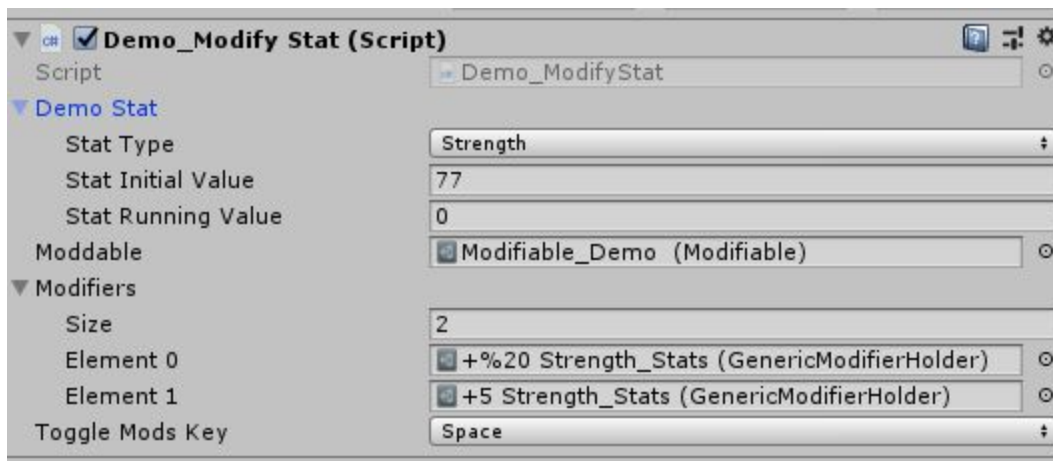
Demo Scene

Includes examples of how to use the **ModifierHolder** and **Modifiable** Scriptable Objects.

1. Open the ‘Demo’ scene.
2. Enter Playmode
3. Click on the buttons and/or hit spacebar.
4. Edit the scripts to see how they are created (to learn how to implement into your own game).

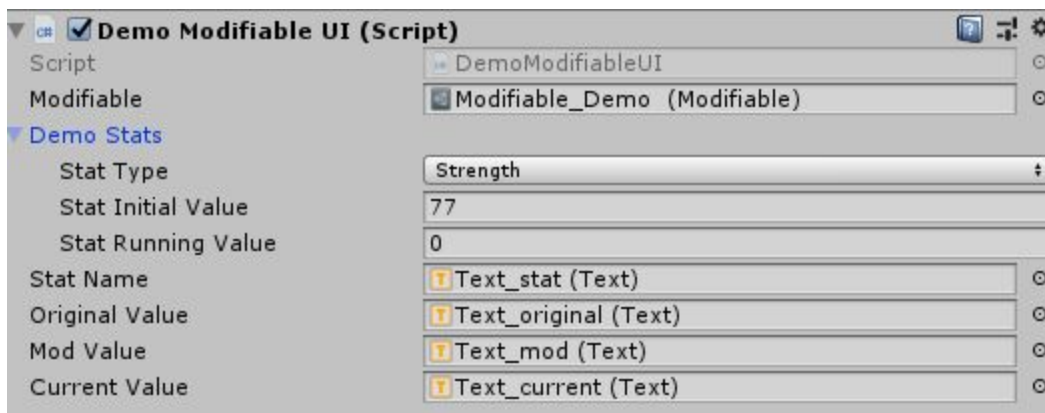
On the **ModModifiers** gameobject:

- Example of how to add mods to existing stats.



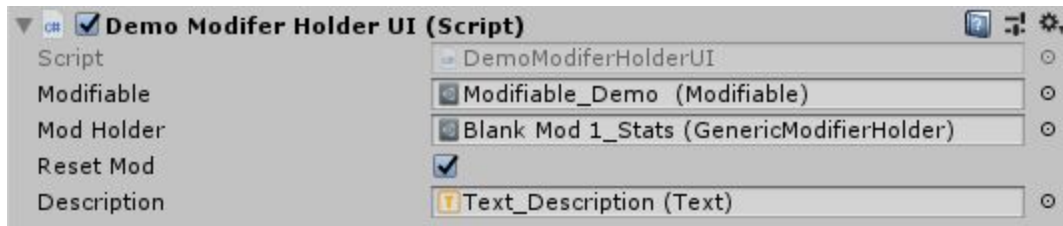
On the **ModCanvas** gameobject:

- Example of how to get and display the mod value, and how to add that value to the current value.



“No Frills” Modifiers

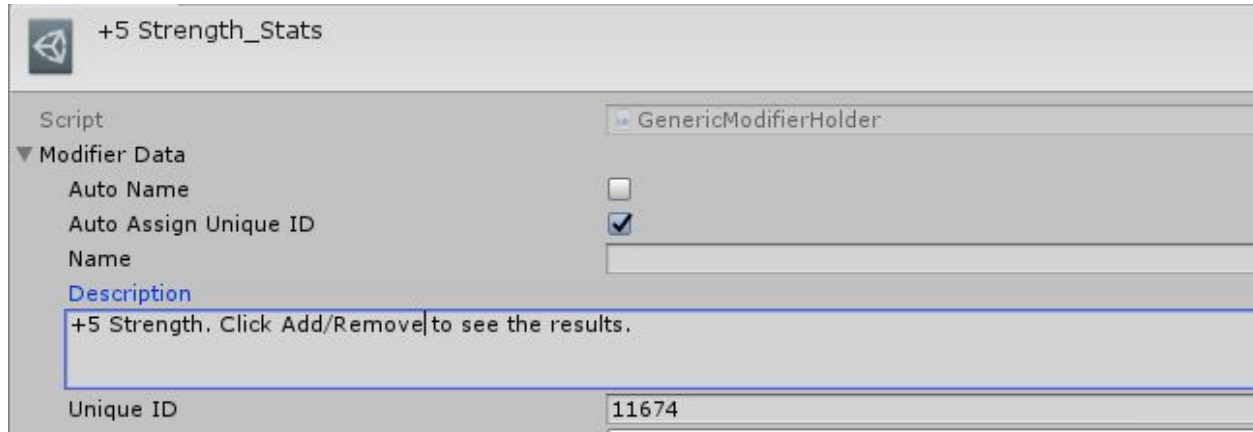
- Example of how to manipulate Base Mod Holders at runtime.



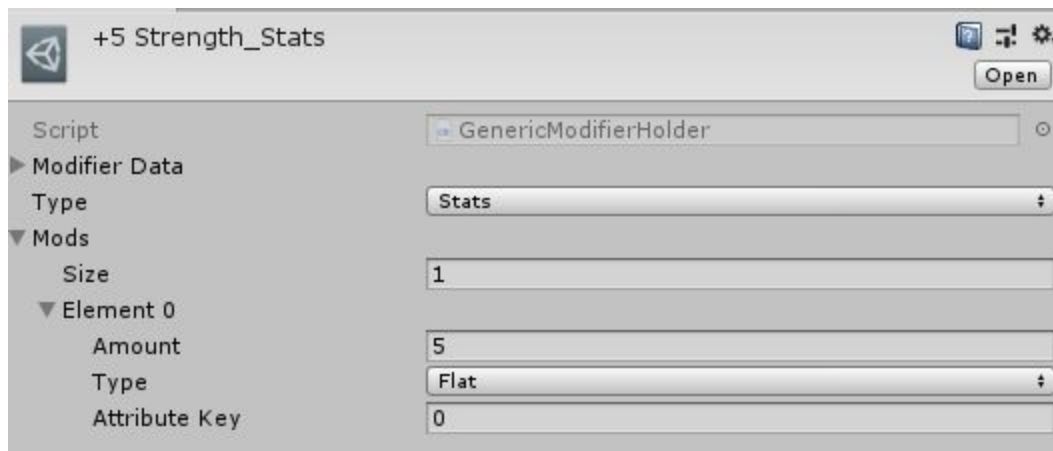
“No Frills” Modifiers

BaseModHolder

The class “BaseModHolder” is abstract, so the pictures here are on the included “GenericModifierHolder”. These will work without any extra coding on your part.



- *Auto Name* - Allow the system to rename the Scriptable Object to match “Name”.
- *Auto Assign ID* - Allow the system to create a *Unique ID* from the instance ID.
- *Name* - The Descriptive name for the Aura.
- *Description* - The Description for the Aura.
- *Unique ID* - The unique ID, used for saving purposes (saving not included in the package).

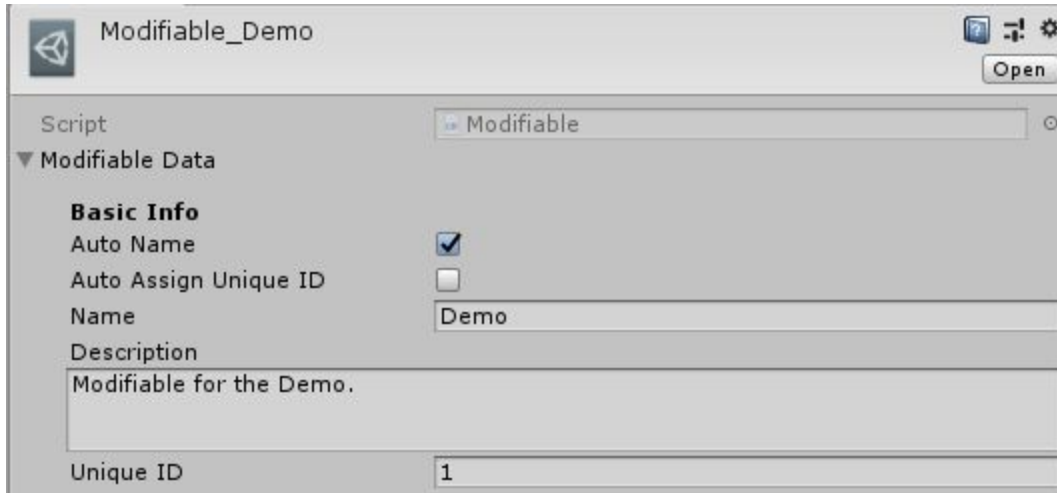


- *Mods* - How many mods does this hold?
- *Amount* - The amount.
- *Type* - Flat or percent modifier. If percent, the amount above is converted to %.
- *Attribute Key* - The attribute to modify. This can be overridden to fit in your own enum, see the Demo Modifier example.

“No Frills” Modifiers

Modifiable

The Modifiable is the thing that remembers which modifiers are currently applied. You can call into it to get your particular mod value an attribute.



The screenshot shows a window titled "Modifiable_Demo" with a toolbar containing icons for a file, a list, and a settings gear, along with an "Open" button. The main area is divided into a "Script" section with a dropdown menu set to "Modifiable" and a "Modifiable Data" section. The "Modifiable Data" section is expanded to show "Basic Info" with the following fields: "Auto Name" (checked checkbox), "Auto Assign Unique ID" (unchecked checkbox), "Name" (text field with "Demo"), "Description" (text area with "Modifiable for the Demo."), and "Unique ID" (text field with "1").

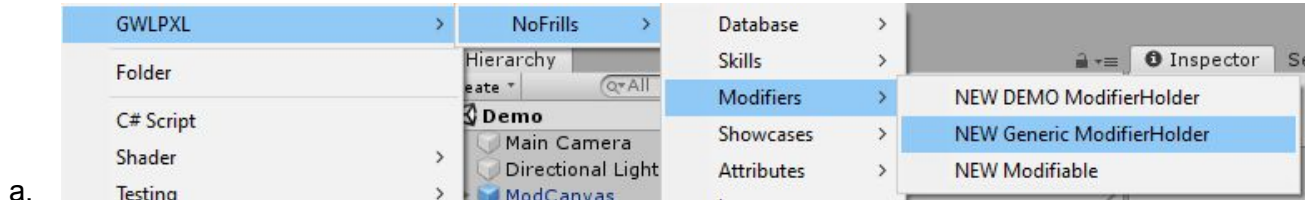
- *Auto Name* - Allow the system to rename the Scriptable Object to match “Name”.
- *Auto Assign ID* - Allow the system to create a *Unique ID* from the instance ID.
- *Name* - The Descriptive name for the Aura.
- *Description* - The Description for the Aura.
- *Unique ID* - The unique ID, used for saving purposes (saving not included in the package).

“No Frills” Modifiers

Create Generic Mod Holder

To create a new one, open up the context menu by right clicking in the *Project Window*.

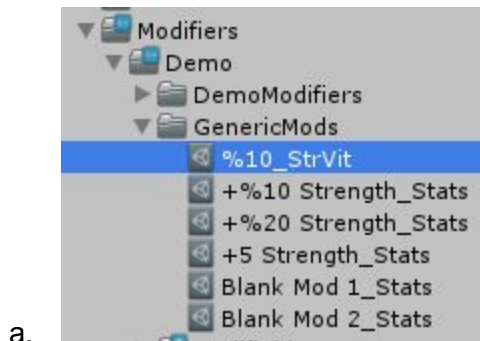
1. Right Click -> Create -> GWLPXL -> NoFrills -> Modifiers > NEW Generic ModifierHolder



2. Re-name the newly created **Generic Mod Holder**.

Or

1. Select an existing **Generic Mod Holder** Scriptable Object.
2. Duplicate it with (Ctrl + D).



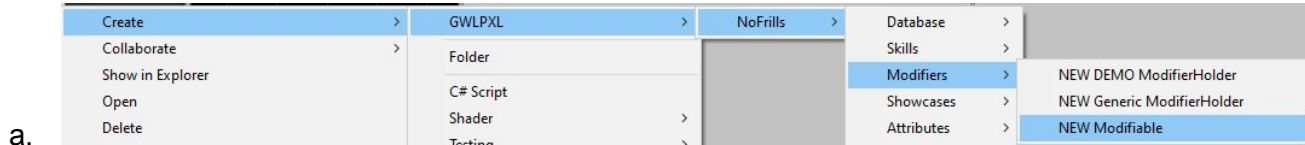
3. Re-name the newly created **Generic Mod Holder**.

“No Frills” Modifiers

Create Modifiable

To create a completely new one, open up the context menu by right clicking in the *Project Window*.

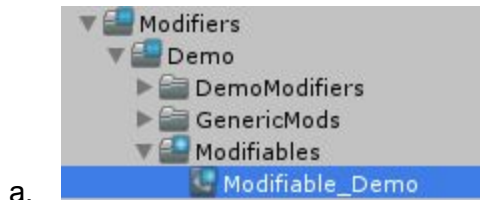
3. Right Click -> Create -> GWLPXL -> NoFrills -> Modifiers > NEW Modifiable



4. Re-name the newly created **Modifiable**.

Or

4. Select an existing **Modifiable** Scriptable Object.
5. Duplicate it with (Ctrl + D).



6. Re-name the newly created **Aura Controller**.

“No Frills” Modifiers

Extending the Package

The following section describes how you can implement the **Modifiers** into your own system and how you can extend what’s already inside.

“No Frills” Modifiers

Create New from ModBase

The “Demo_Modifier” script shows you how to extend modifiers so they display your own enum without having to remember the int keys.

```
/// <summary>
/// Demo example of how to extend the modifiers to your own system.
/// </summary>
[System.Serializable]
1 reference
public class Demo_Modifier : ModBase
{
    //insert your own Enum here.
    public DemoStats Stat;

    5 references
    public override int GetAttributeToModify()
    {
        return (int)Stat;
    }

    4 references
    public override void SetAttributeToModify(int enumKey)
    {
        Stat = (DemoStats)enumKey;
    }
}
```

“No Frills” Modifiers

Create New From BaseModHolder

The “Demo_ModifierHolder” shows you how you can then create a unique holder for your newly created Modifiers.

```
// </summary>
[CreateAssetMenu(menuName = "GWLFXL/NoFrills/Modifiers/NEW DEMO ModifierHolder")]
public class Demo_ModifierHolder : BaseModHolder
{
    [SerializeField]
    ModifierType Type = ModifierType.Stats; //we must define a type.
    [SerializeField]
    Demo_Modifier[] DemoMods; //use the base class you created from ModBase.

    //must override and return our new class
    9 references
    public override ModBase[] GetAllModifiers()
    {
        return DemoMods;
    }

    //must return modifier type
    7 references
    public override ModifierType GetModType()
    {
        return Type;
    }
}
```

“No Frills” Modifiers

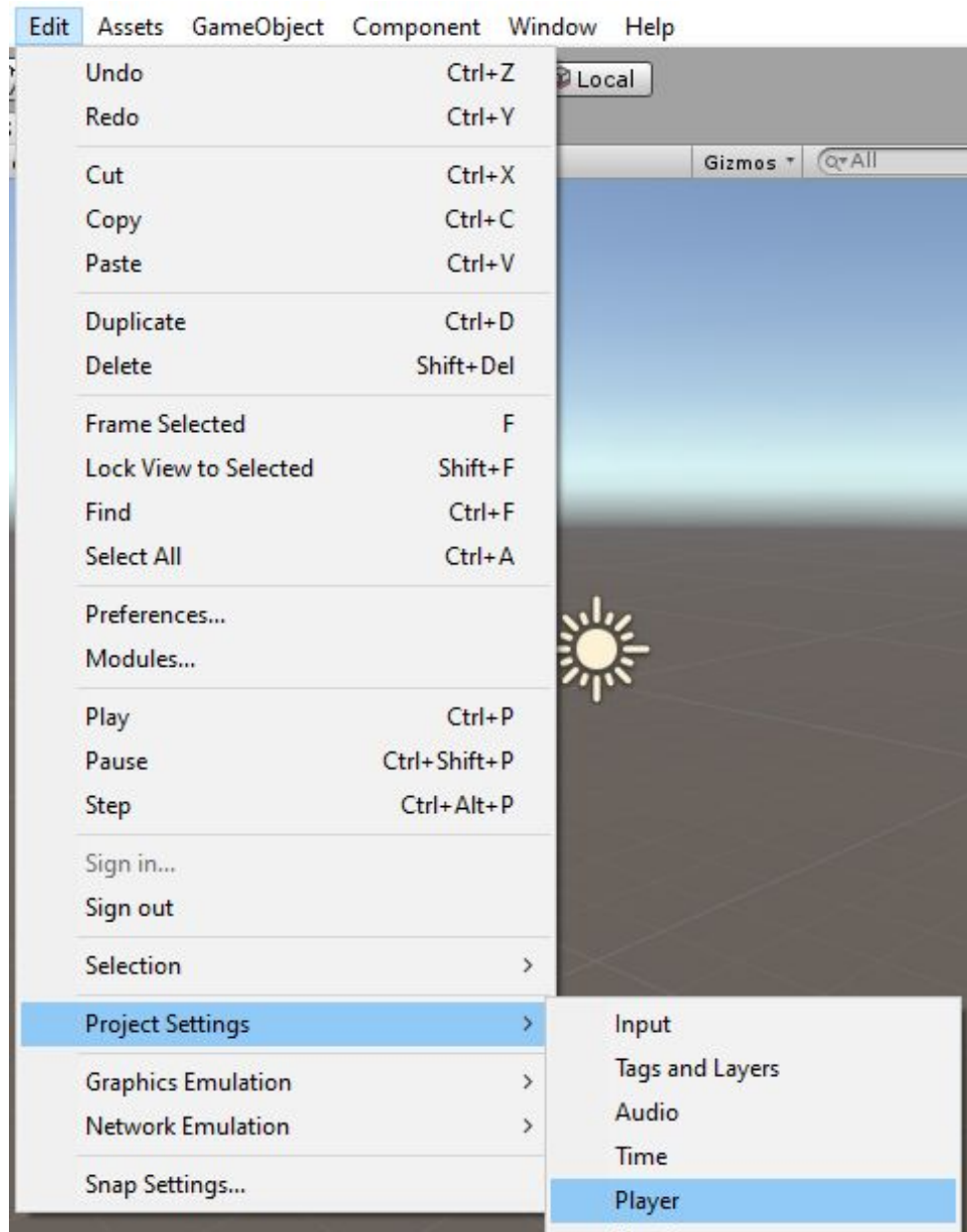
F.A.Q.

Question: Help! I’m getting this error!

error CS1644: Feature 'declaration expression' cannot be used because it is not part of the C# 4.0 language specification

Answer: You need to be on **Unity 2018** and **.NET 4.X**. Follow the directions below to upgrade your .Net (you will need Unity 2018 or higher in order to do so).

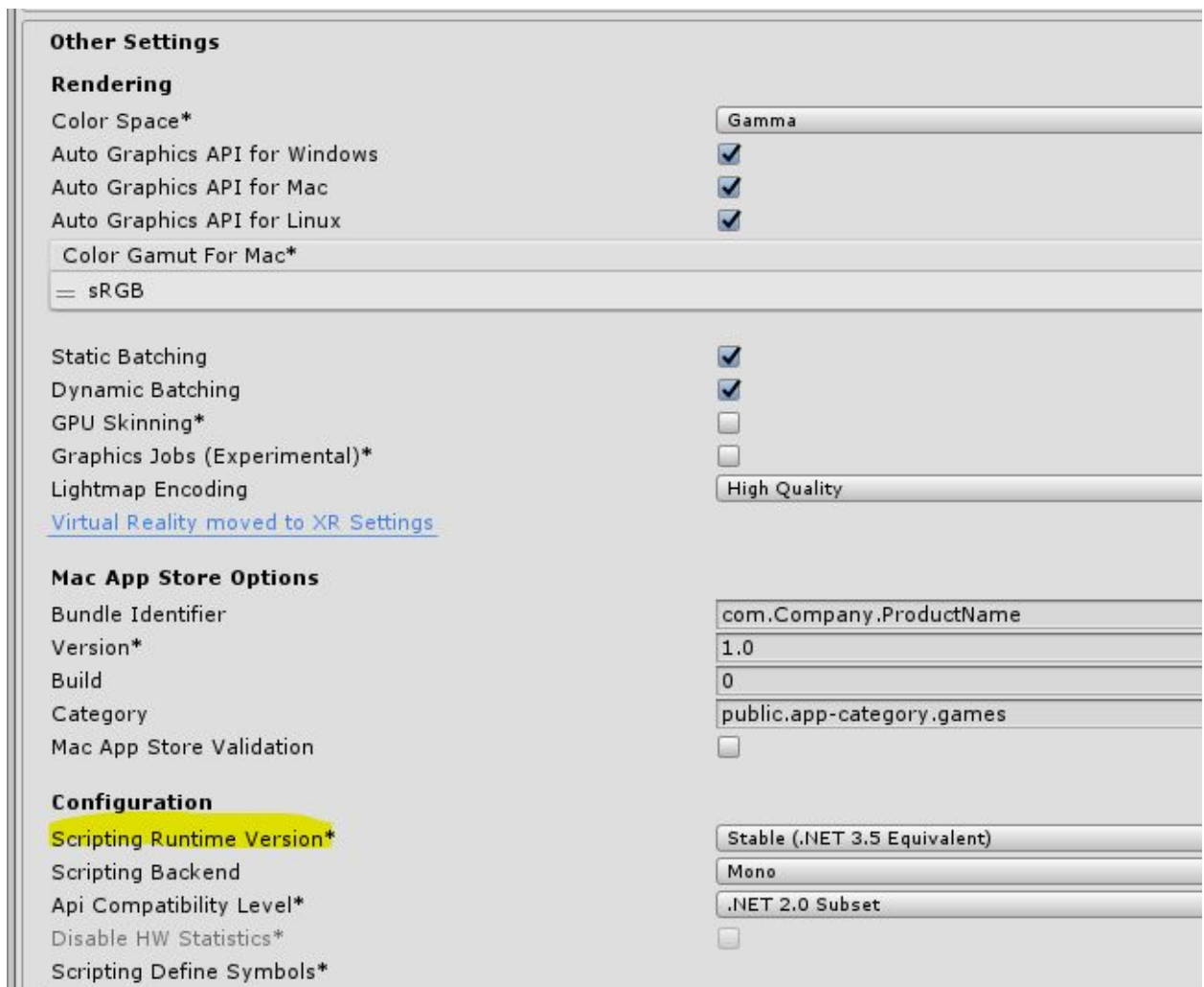
1. Navigate to Edit -> Project Settings -> Player



- 2.

“No Frills” Modifiers

3. Navigate to Other Settings, and click to fold it out.



- 4.
- 5.
6. Locate Scripting Runtime Version and make sure it's .NET 4.x Equivalent or higher.



- 7.
8. Save.
9. Exit and Re-open the project.
10. Done.