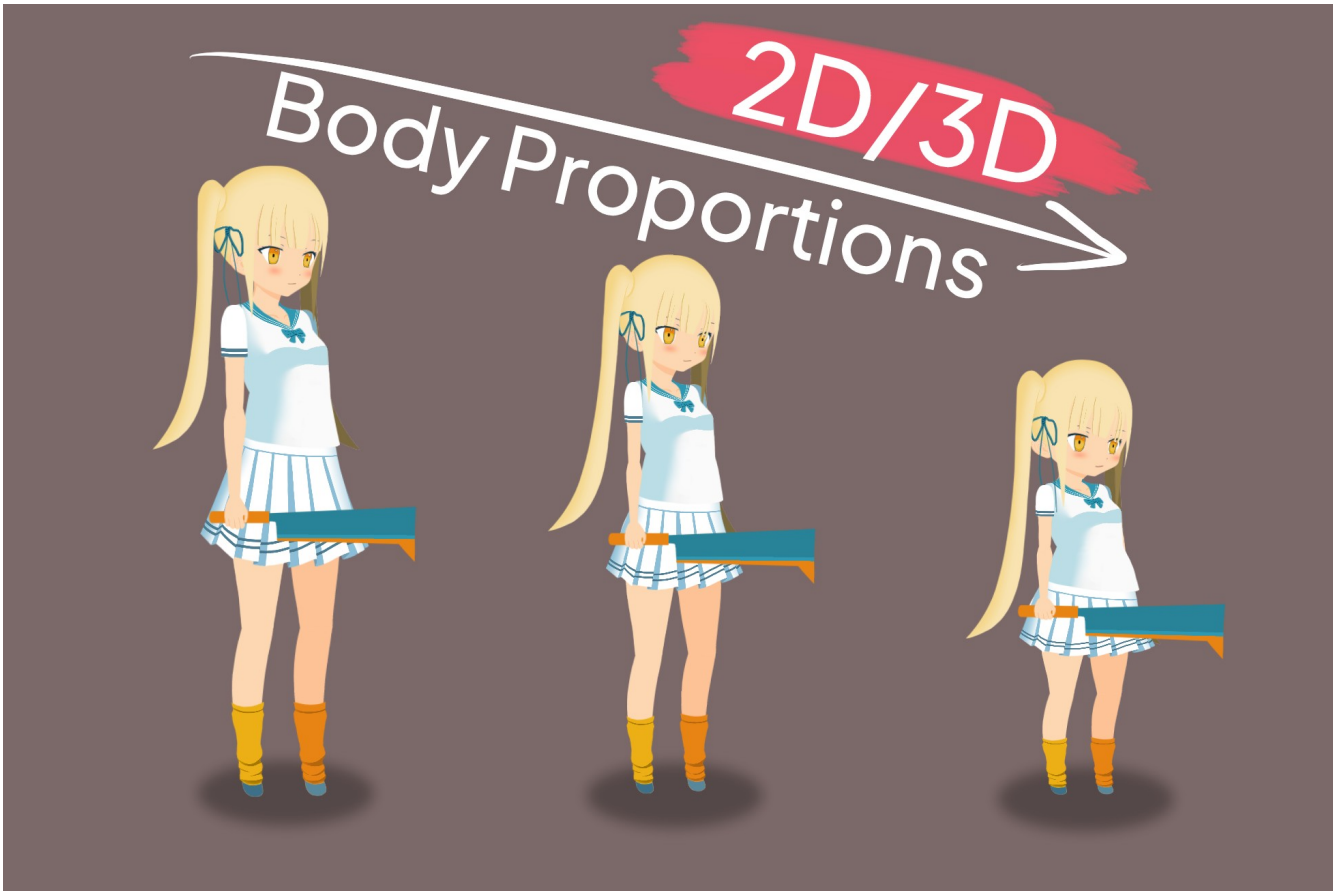
Body Proportions

User Guide



Catalog

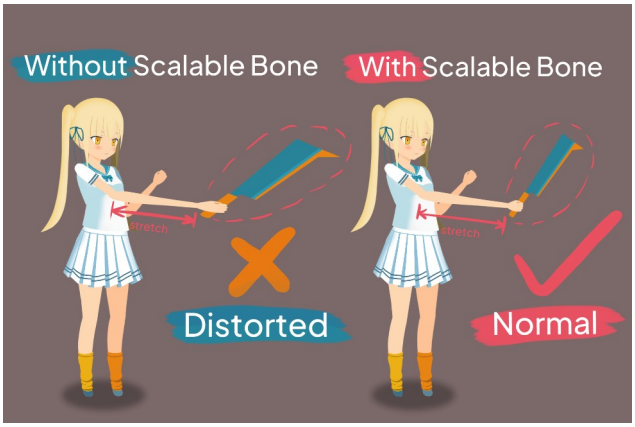
1 Introduction

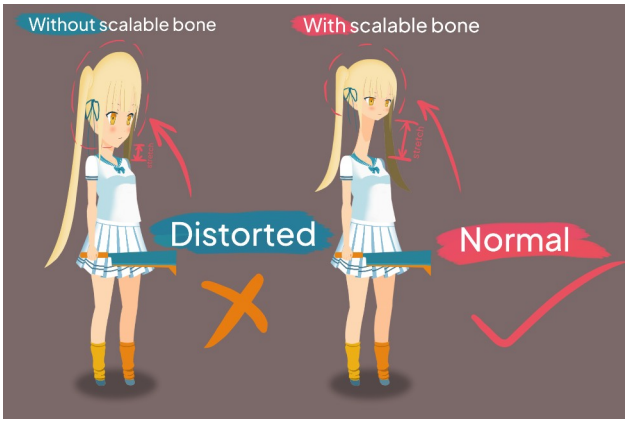
Body Proportions is a tool that gives you the ability to change the proportions of the model's body by scaling the bones. You can use it to make animations as usual.

With Body Proportions you can easily change the length of your character's body parts and create more characters without more materials.

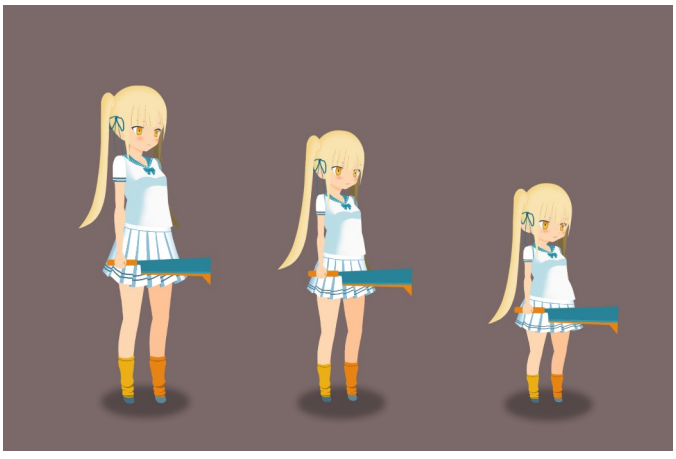
How it work?

Usually the skeletal structures of a model are nested within each other, so when you move the parent skeleton, the child skeletons will move with it. For example, when a rotating arm bone lifts up, the hand bone lifts up as well. It's very easy to use nested bone structures to handle this kind of movement and rotation, but it's not possible to maintain the normal effect when the bones are scaled, e.g. if you want a longer leg or arm, the hand will be deformed.





In the past, you had to re-edit the mesh using blender or other model editing software. With Body Proportions, you can easily make changes by simply adjusting the scale property in Transform.
This is because Body Proportions breaks up the skeletal structure, puts the bones at the same level, and then re-associates them for easier manipulation.
You can use Body Proportions on your model as long as the skeleton is visible in the hierarchy and the bones affect the model in a tree-like pattern, Body Proportions supports 3d models as well as 2d characters created by unity 2d animation.



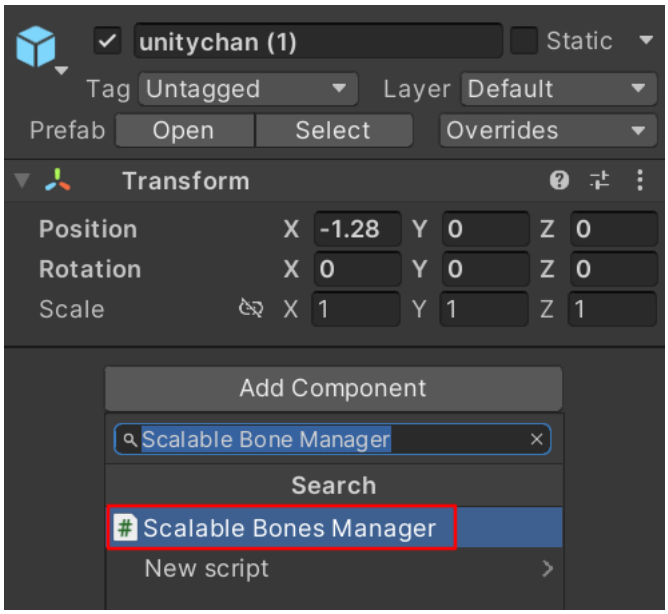
2 Getting Started

In the tutorials we use Unity-Chan as a demo model, thanks to UNITY for allowing us to use her as a demo model, if you want to use her in your project, please check the license under /Body Proportions/unity-chan!/Documentation.
It’s recommended that open the scene under /Body Proportions/Scenes/Tutorials to learn how to use the tool interactively with the explanations in this manual.
Let's get started!

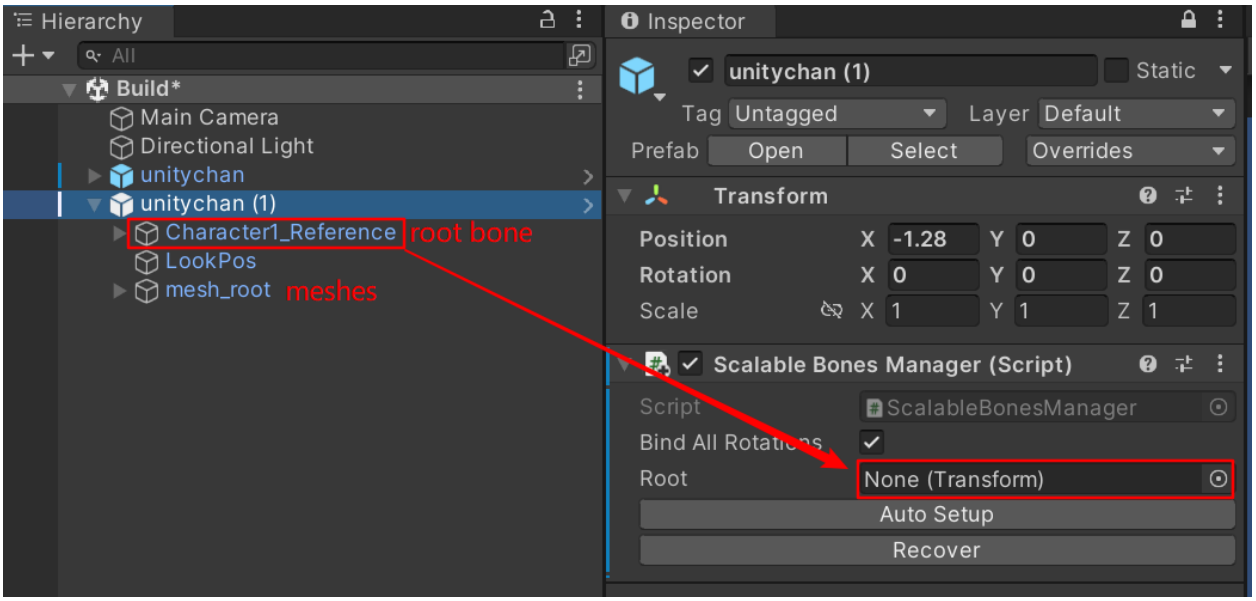
2.1 Setup Body Proportions to your model

1.Backup your model before everything. *ScalableBonesManager* may overwrite your gameObject.

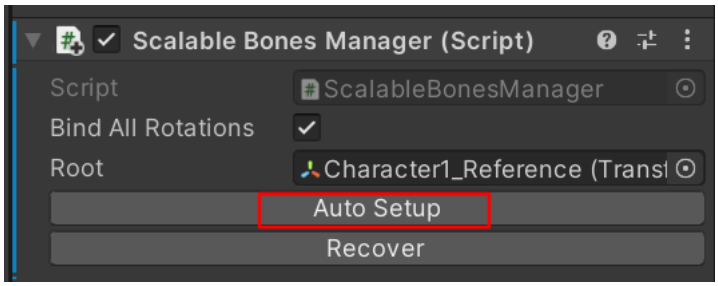
2.Select your model instance and add the component *ScalableBonesManager*.



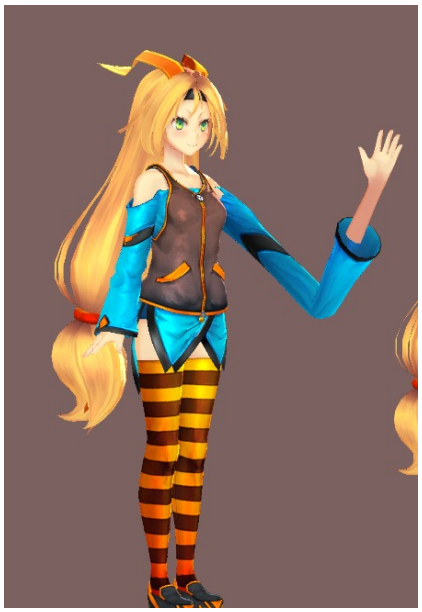
3.Specify the root skeleton.



4. click the "auto setup" button



5. Now try scaling the arm bones of the model.



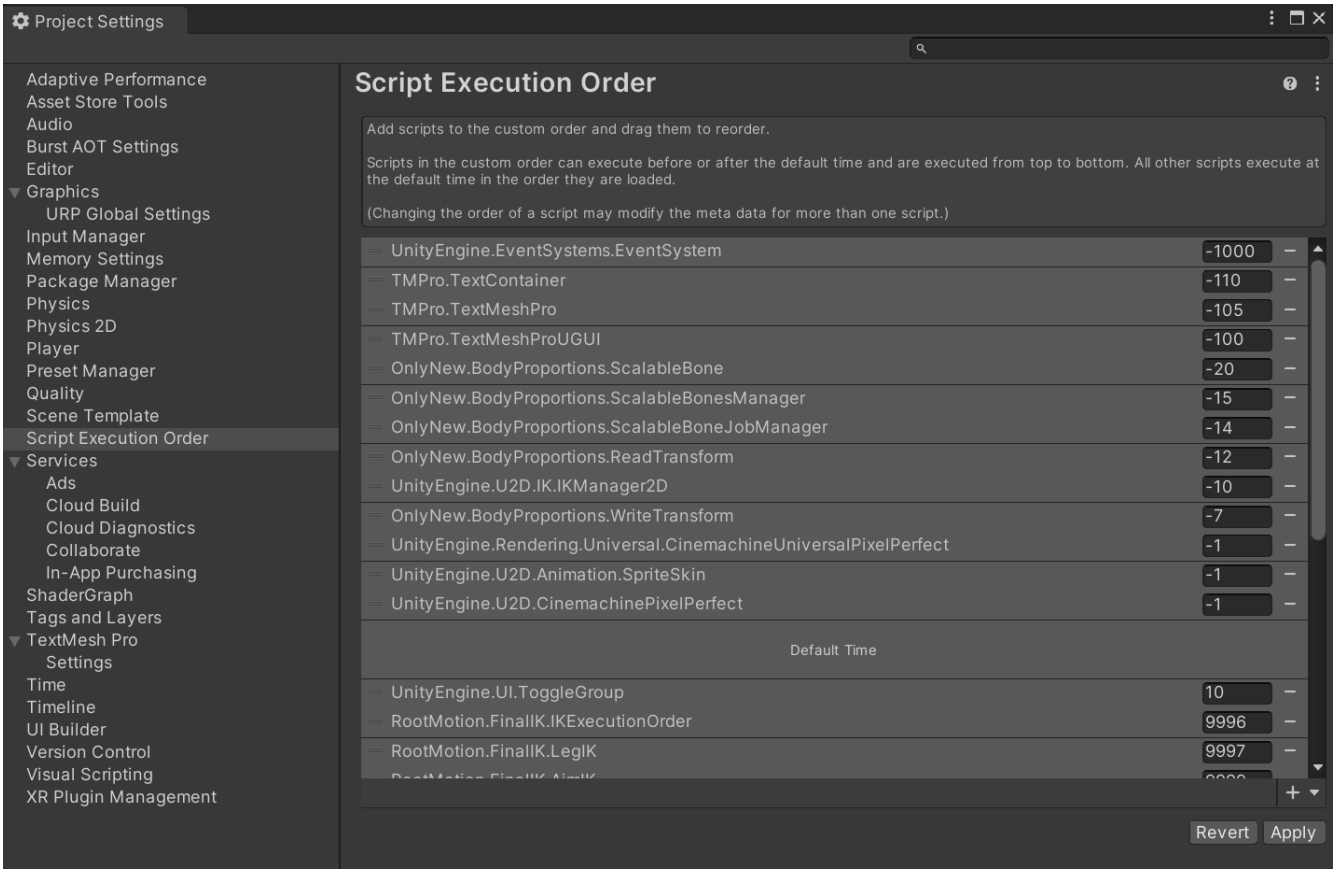
2.2 Using IK with Virtual Skeleton

This asset supports Final IK by default. if you are using other IK, you will need to change some settings. At the end of this chapter, you'll find configuration examples for several popular IK methods, such as Final IK, Bio IK, Unity 2D IK and Animation Rigging. **Even if the IK you are using is not listed, you can still make it work by applying the underlying theory.**

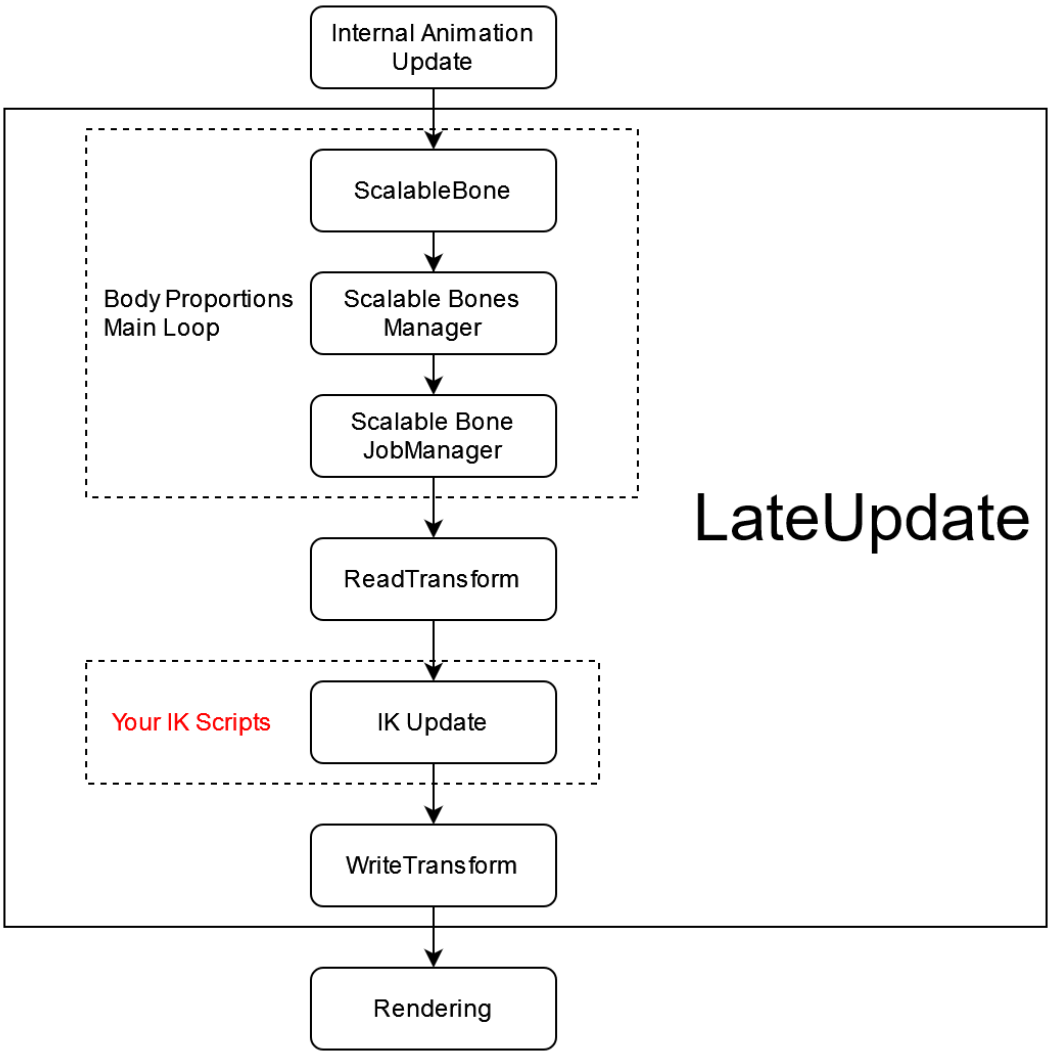
2.2.1 Theory

Since most IK systems are designed for coupled skeletons, they cannot function correctly on decoupled skeletons. To ensure these IK scripts work properly, we need to create a familiar environment for them. We will generate a coupled virtual skeleton, similar to the one you typically use.

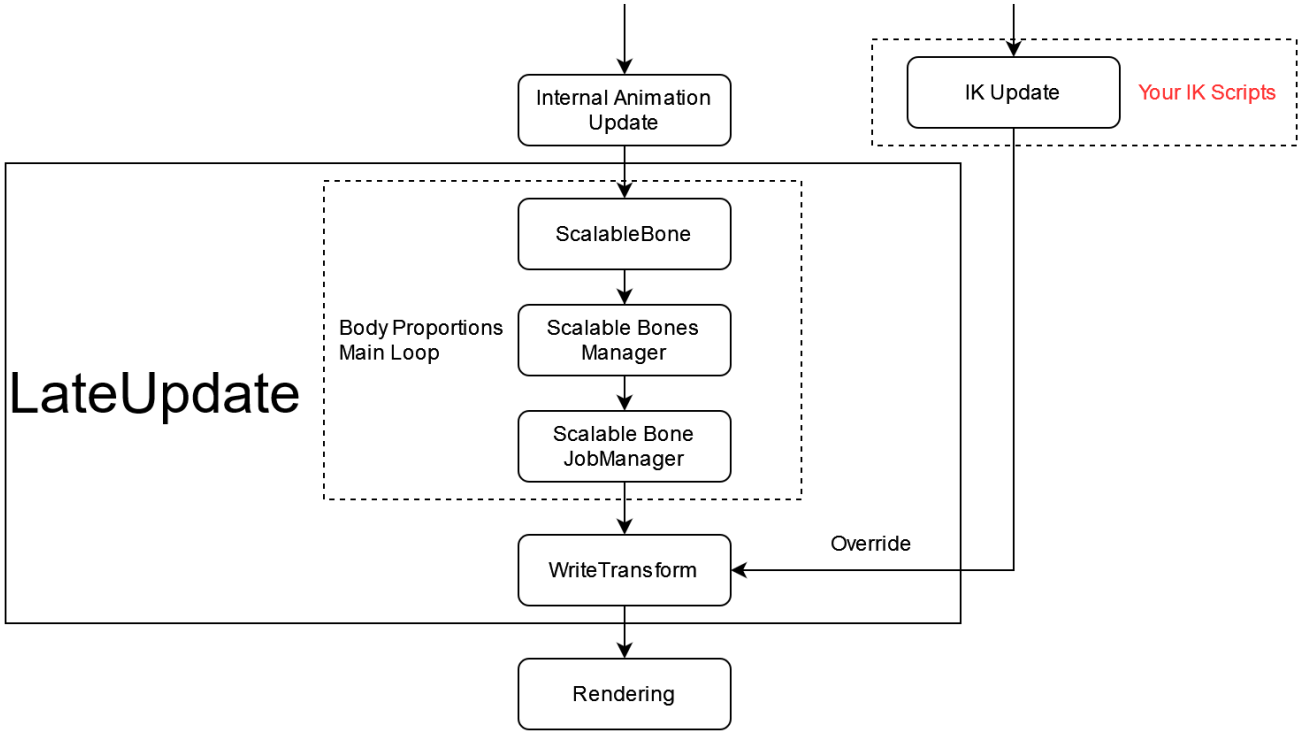
Before the IK update, we will read the position and rotation data from the real skeleton and transfer it to the virtual skeleton. The IK script will then solve it on the virtual skeleton. After the IK update, the position and rotation data from the virtual skeleton will be written back to the real skeleton. To implement this process, we need to **modify script execution order**.



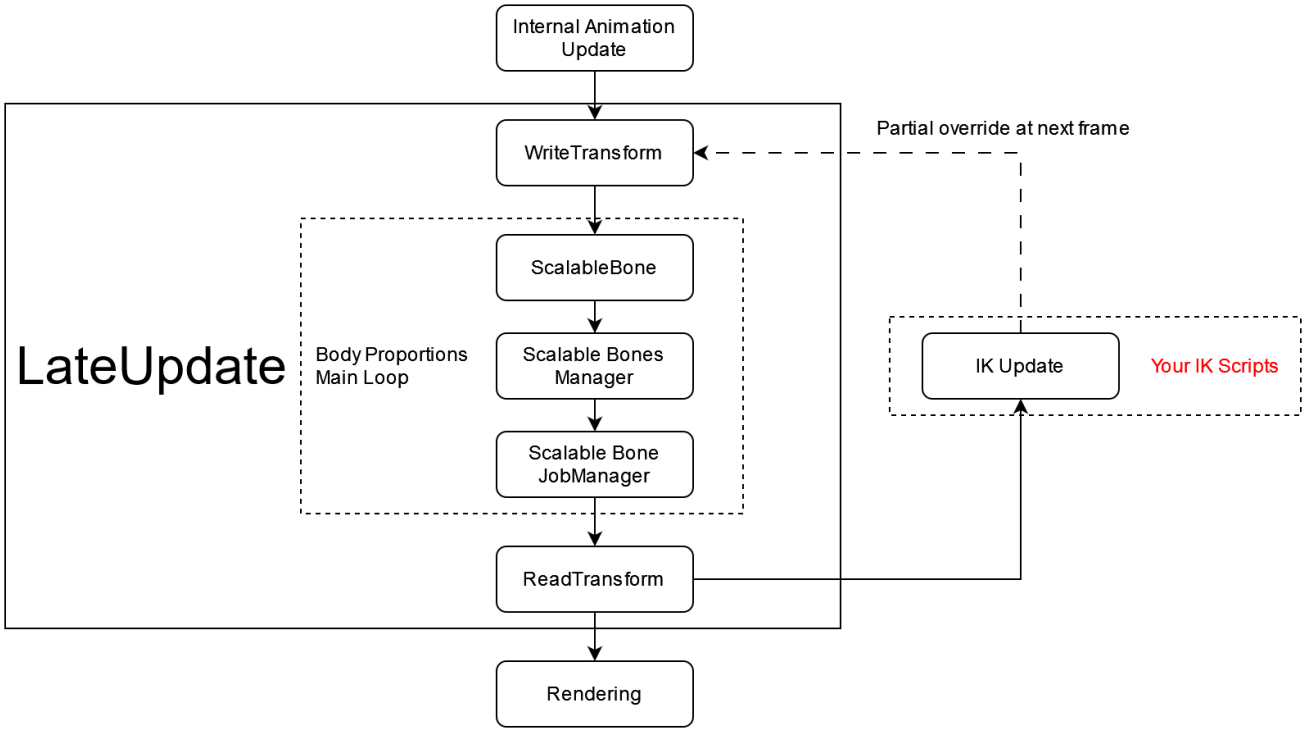
Both Scalable Bone and IK work properly when the execution order of the scripts matches the flow shown below.



If your IK scripts are not solved in LateUpdate(), or if you are unable to modify the execution order of the IK scripts, then we need to follow another script execution order. In this order, IK scripts will override the animation and cannot interpolate between the solving result and the animation.



If you don't mind the IK calculations lagging by one frame, we can use a third method. Its advantage over the second method is that you can dynamically change the body proportions during runtime.



2.2.2 Configuration Steps

Before you begin, please familiarize yourself with the standard usage of your IK to avoid misidentifying the source of errors.

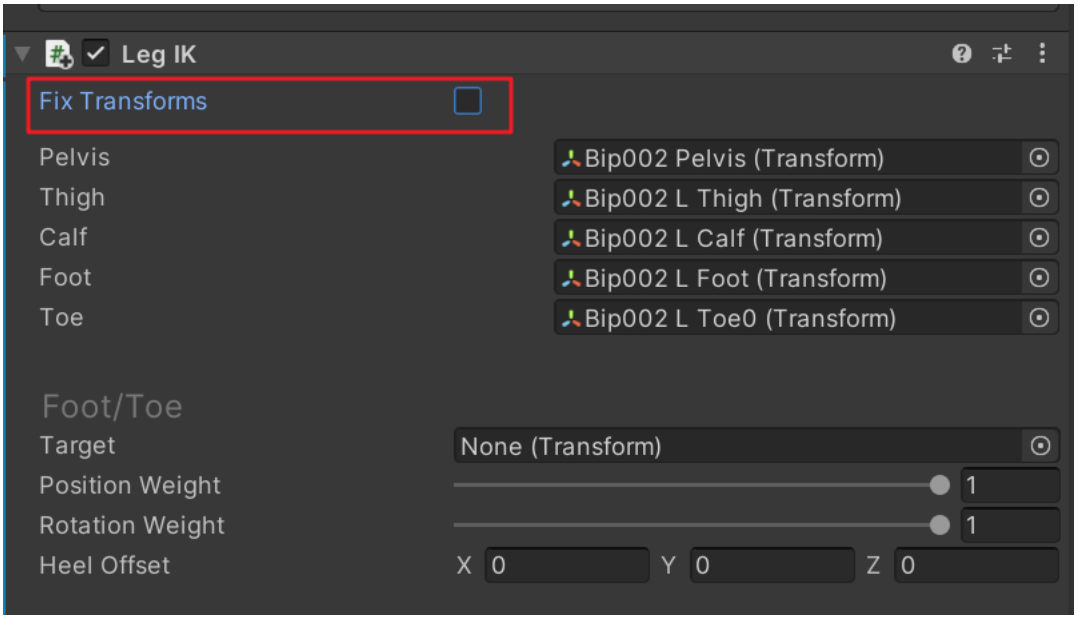
Before proceeding, please setup Scalable Bone and create a virtual skeleton.

Final IK

1. Install your IK onto the virtual skeleton.

2. As this asset's script default execution order is compatible with Final IK, no modifications are required.

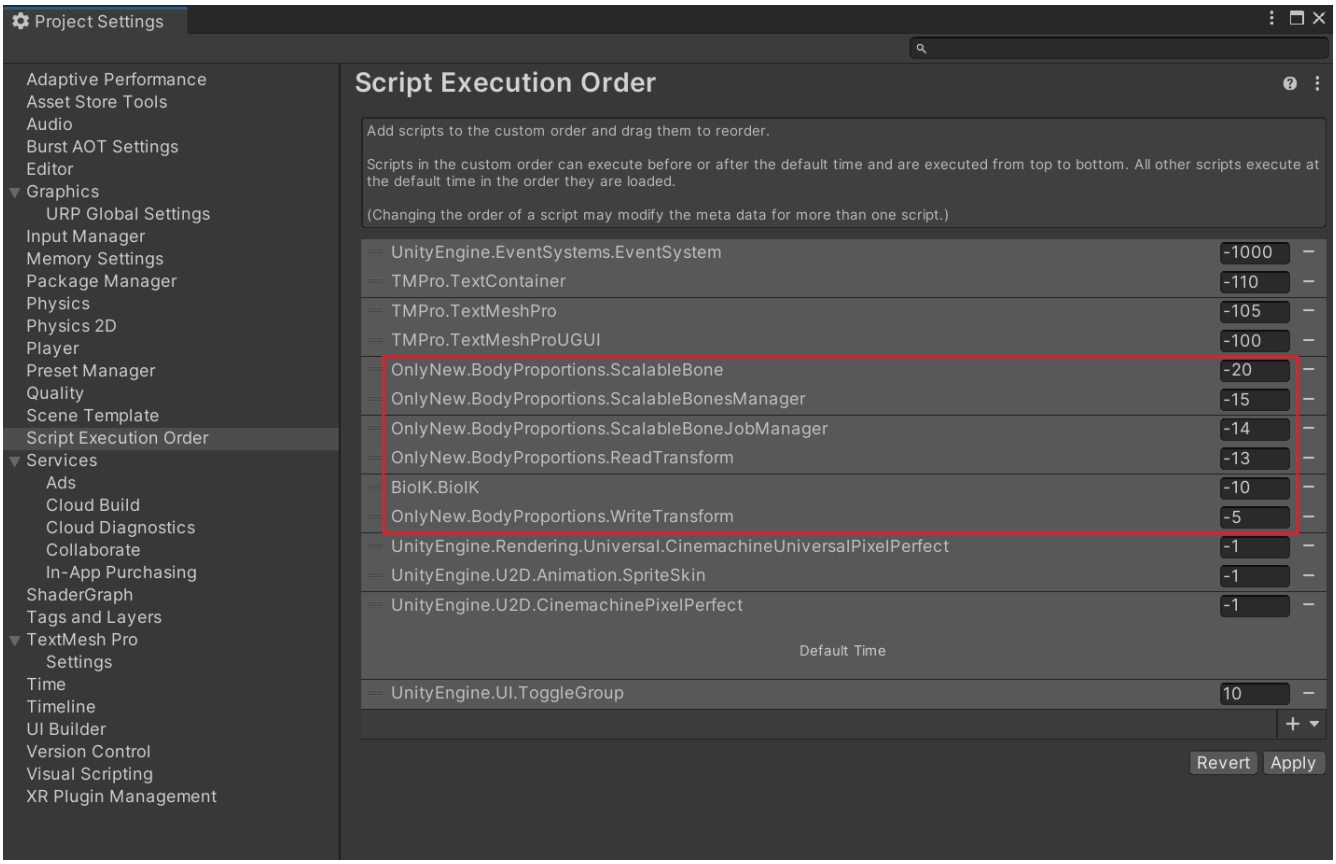
3. Set "Fixed Transform" to false on Final IK inspector.



Bio IK

1. Install your IK onto the virtual skeleton.

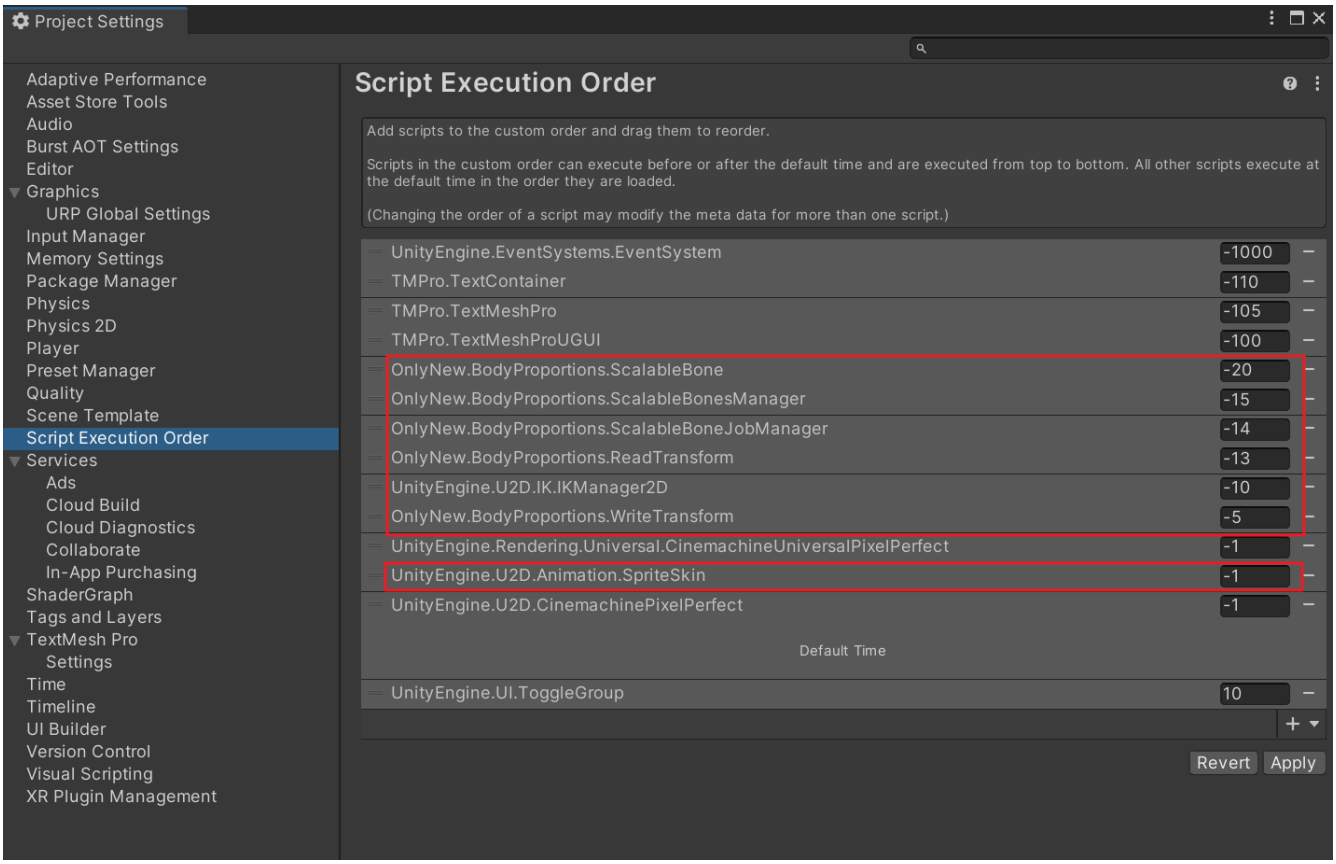
2. Modify the script execution order as shown in the following picture.



Unity 2D IK

1. Install your IK onto the virtual skeleton.

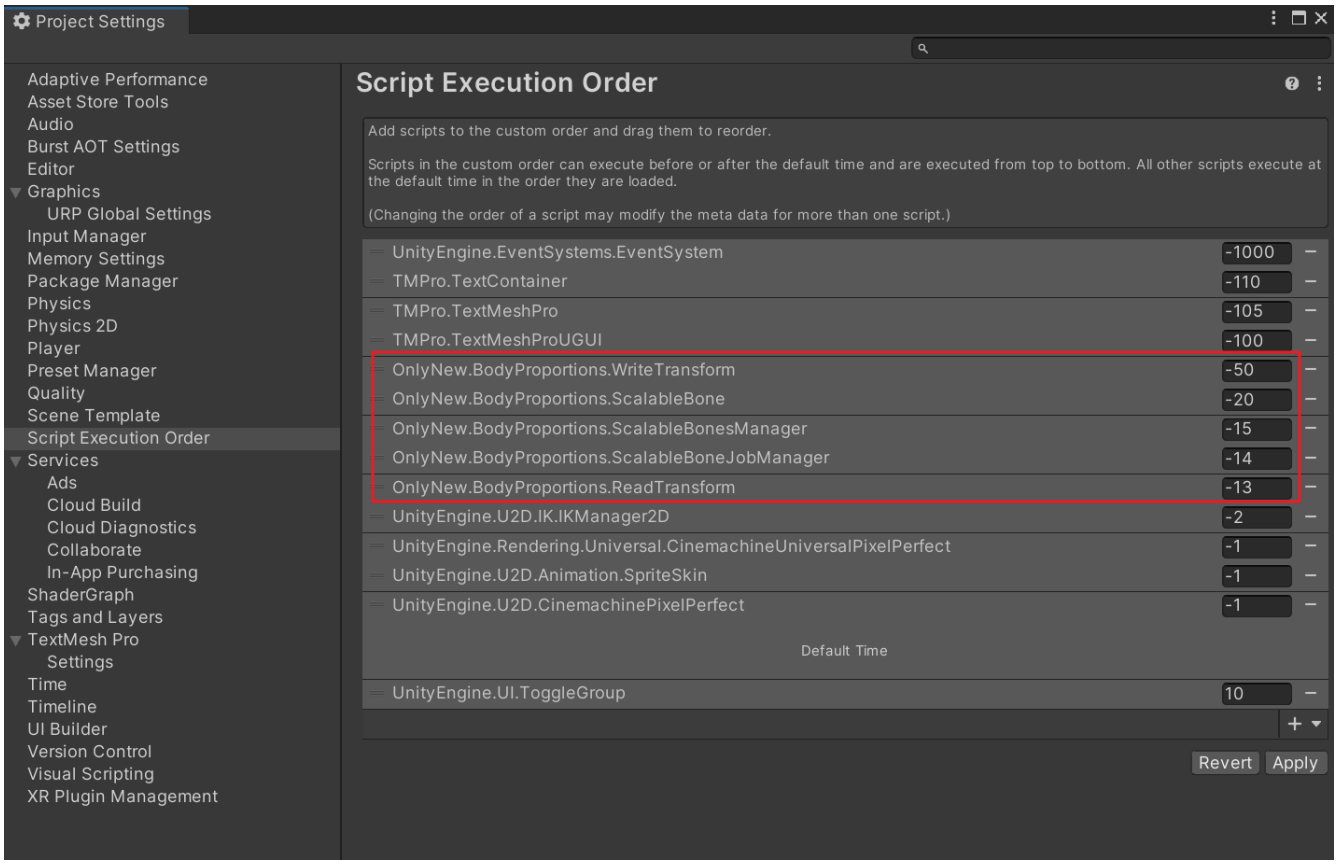
2. Modify the script execution order as shown in the following picture.



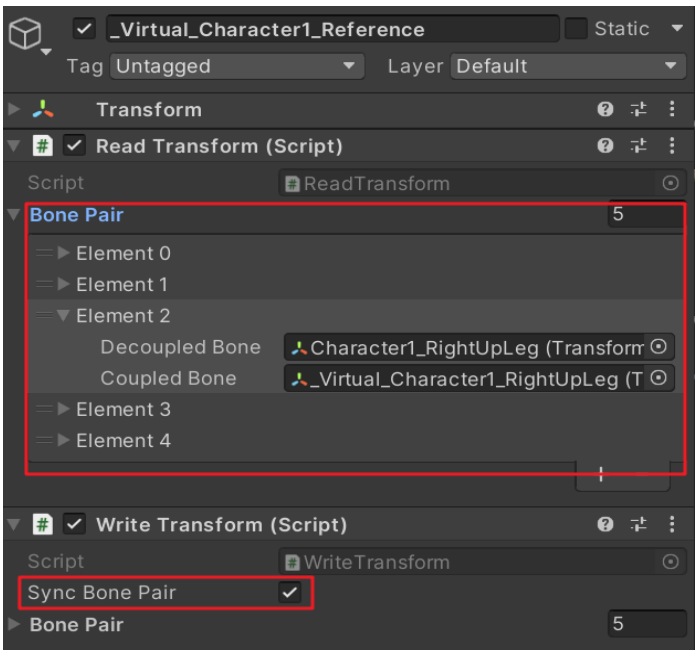
Unity Animation Rigging

1. Install your IK onto the virtual skeleton.

2. Modify the script execution order as shown in the following picture.



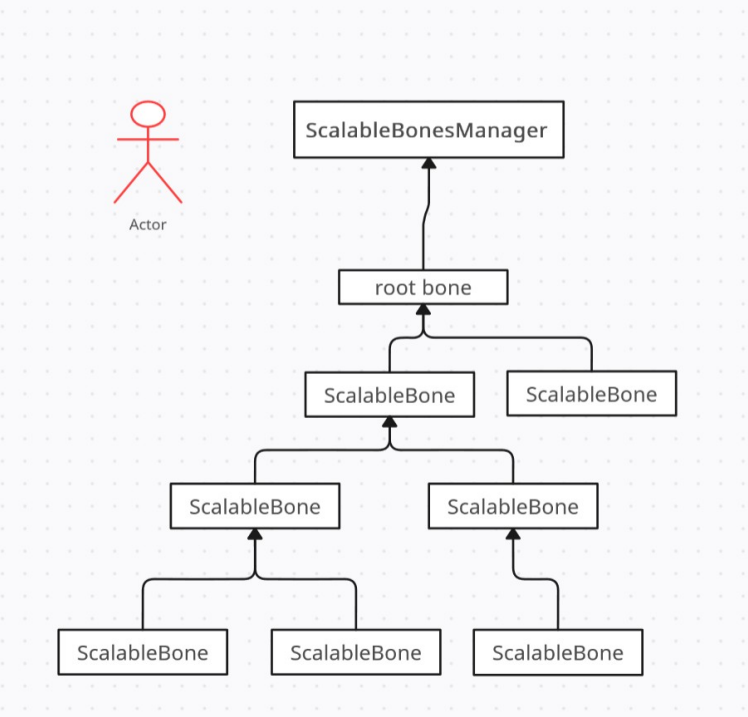
3.In the inspector of ReadTransform, clear the bonePair and add the bones covered by IK. In the inpector of WriteTransform, Set "SyncBonePair" to true.



3.Manual

3.1 ScalableBone

ScalableBones work together. Each *ScalableBone* moves and rotates with the parentScalableBone, but does not scale with the parentScalableBone.



field	type	description
parentScalableBone	ScalableBone	The bone which it follows. Usually it is configured by ScalableBonesManager. In the future it will support manual configuration.
bindPosition	bool	If it is true, It moves with its parentScalableBone.
bindRotation	bool	If it is true, It rotates with its parentScalableBone.
positionOffset	Vector3	The offset that is maintained when moving with its parentScalableBone.
debug	bool	If it is true, print logs about this bone.

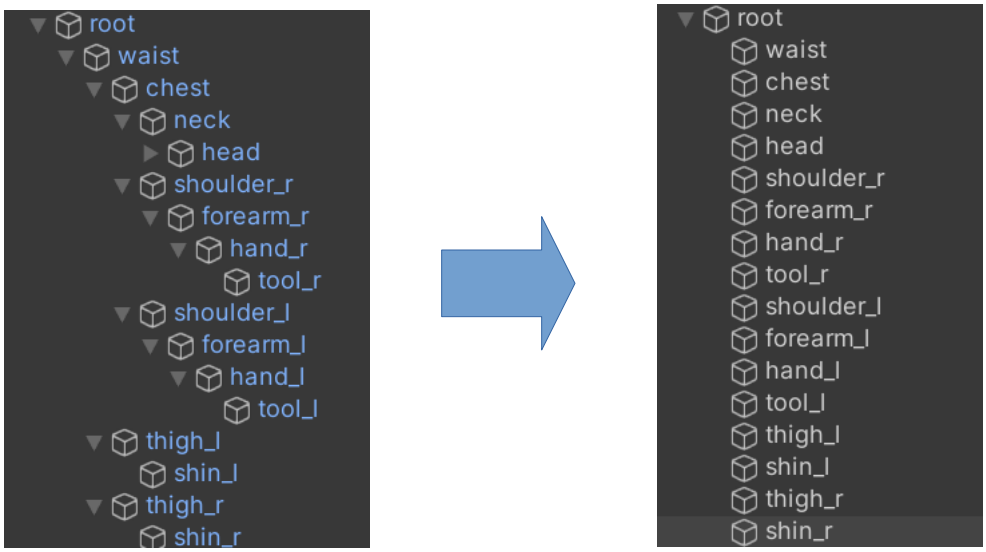
3.2 ScalableBonesManager

The *ScalableBonesManager* is used to manage the installation and uninstallation of ScalableBones. It is also used to collect and execute events in the *ScalableBone*.

field	type	description
bindAllPositions	bool	When it changes, the bindPosition of all the children *ScalableBone* of root changes.
bindAllRotations	bool	When it changes, the bindRotation of all the children *ScalableBone* of root changes.
root	Transform	After clicking the "Auto Setup" button, all child bones of root will be configured automatically.
Multi-threaded Mode	bool	Enabling multi-threaded mode can make the program faster when there are many characters in a scene.

3.2.1 Auto Setup

After clicking the "Auto Setup" button, the *ScalableBonesManagerEditor* will reorganise hierarchy of bones, and make root the parent of each child-bone, then adds *ScalableBone* component for each child.

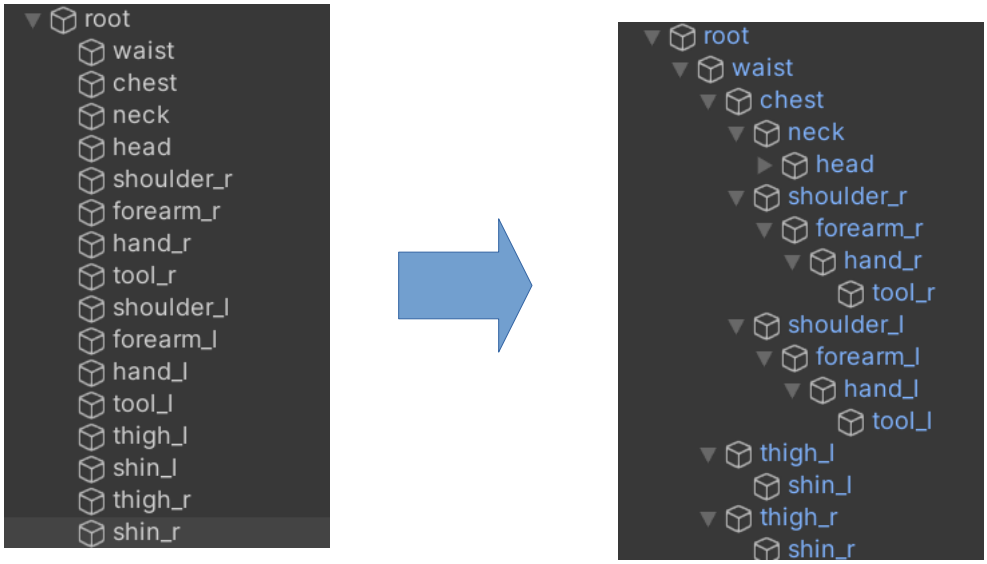


Hierarchy Of Skeleton Before Setup

Hierarchy Of Skeleton After Setup

3.2.2 Recover

After clicking the "Recover" button, the *ScalableBonesManagerEditor* will recover hierarchy of bones, and remove *ScalableBone* components from child-bones.



Hierarchy Of Skeleton Before Recover

Hierarchy Of Skeleton After Recover

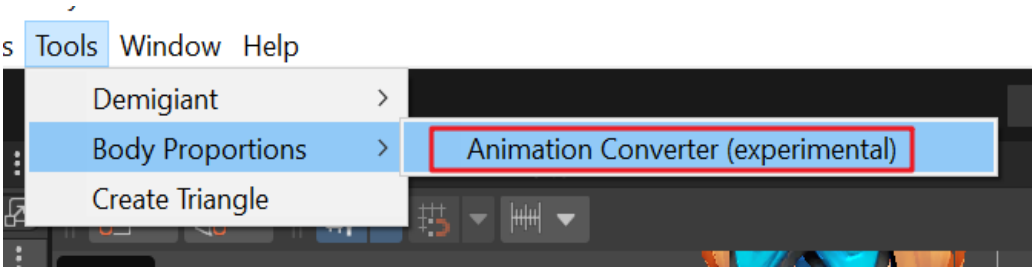
3.3 Scalable Bone Renderer

This component can be used to draw bones. It's self-maintained. you just need to add it.

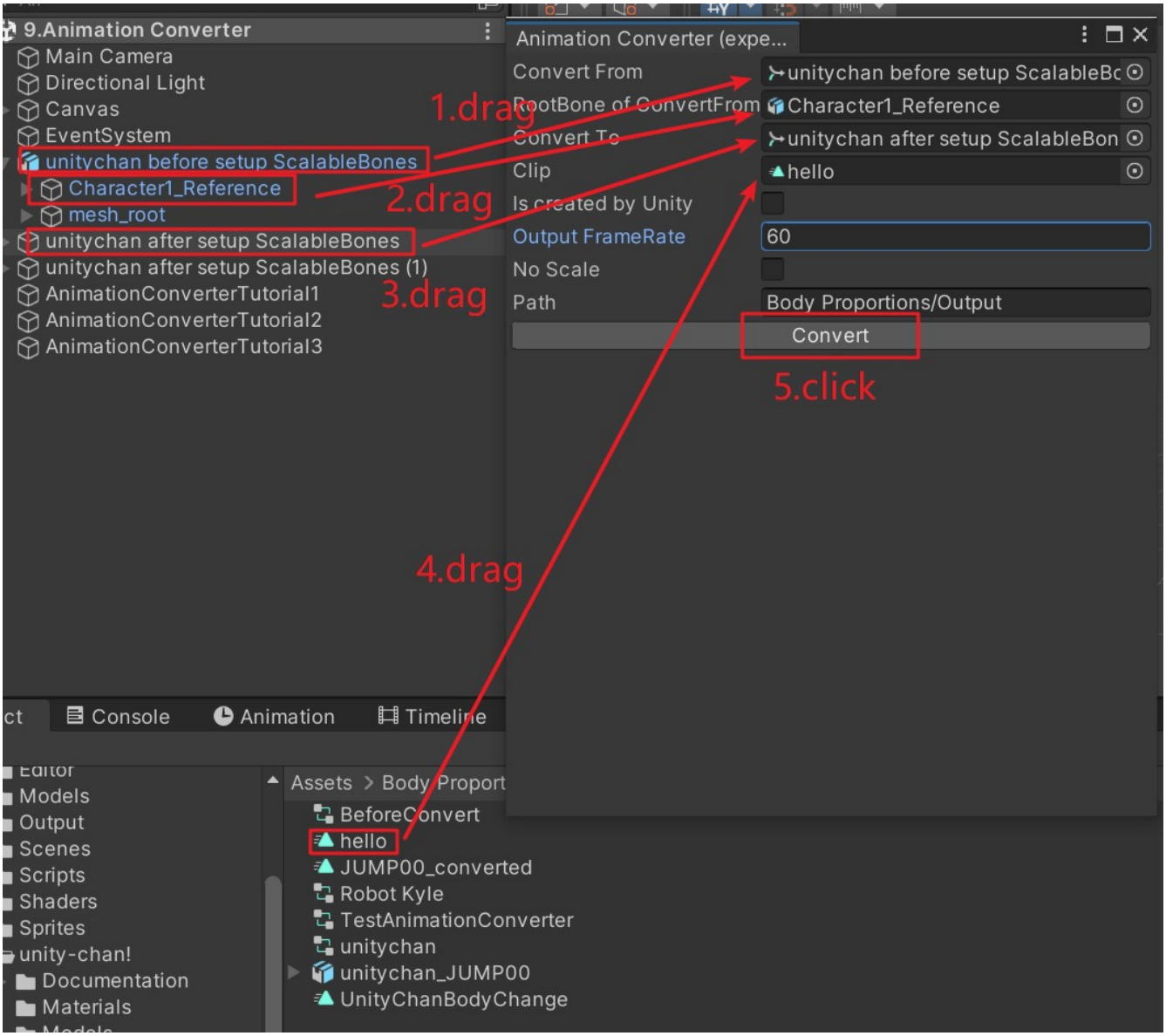
3.4 Animation Converter

Animation Converter is designed to convert animation to new animation which supports Body Proportions. You may find that the previous animation does not work because *ScalableBonesManager* changed the hierarchy of skeleton. If you are experiencing this kind of problem, please use this tool.

You will find it in the path “/Tools/Body proportions/Animation Converter”.



Since **Animation Converter** calls unity's interface, it can support animations that Unity supports by default. Since I have limited animation types, I have only tested it on a small scale. It is confirmed that **Animation Converter** supports Unity animations and FBX animations. If you find an animation that is not supported, please let us know.



3.3.1 How to use?

Backup your model before converting. **Animation Converter** or *ScalableBonesManager* may overwrite your gameObject.

Prepare a model without *ScalableBonesManager*. Put this model into **"Convert From"**.

Duplicate this model and install *ScalableBonesManager*. Put the replica into **"Convert To"**.

Put the animation clip you want to convert into **"Path"**. Then click Button **"Convert"**.

If the toggle **"Is created by Unity"** is unchecked, **Animation Converter** supports any type of animation it can. If it is checked, **Animation Converter** only supports animation created by Unity Animation or Unity Timeline. If an animation created by unity and you checked "Is created by Unity", Animation Converter will reduce unnecessary animation properties.

If **"No Scale"** is checked, scale animation will be discarded when converting.

You will find the converted animation clip in /Body Proportions/Output.

You can also use AnimationConverter.ConvertClip() to convert animation in scripts.

3.5 Protect Children From ScalableBonesManager

By default, after clicking the "Auto Setup" button, *ScalableBoneManager* automatically converts all bones to Scalable Bones. If a bone has a *ProtectChildrenFromScalableBonesManager* component, its children will not be converted to Scalable Bone.

3.6 ScalableBoneJobManager

If multi-threaded mode is enabled, *ScalableBonesManager* creates *ScalableBoneJobManager* automatically. *ScalableBonesManager* sends data of child bones to *ScalableBoneJobManager* on awake. Once Job Manager receives the data, it rebuilds the job data. The rebuild process is time-consuming, and it is recommended to use **Object Pool** to reduce performance losses. ScalableBoneJobManager has three update modes：

- **For Editor**：This is the default mode in editor, regardless of whether the application is playing or not. It allows you to make animations.
- **For Runtime**: This is the default mode in the separated build version of the game. You can use ScalableBone.SetRotation() or ScalableBone.IsBeingDragged to move the skeleton group.
- **Fast Mode**：You need to enable it manually in inspector. you can’t make animations while application is Playing. ScalableBone.SetRotation() or ScalableBone.IsBeingDragged is unavailable.

4 Troubleshooting

4.1 UnpackPrefabInstance must be called with a root Prefab instance GameObject

Solution: Please unpack the parent Prefab of *ScalableBonesManager*, and try to setup again.

4.2 Unsynchronised movement of child-bones

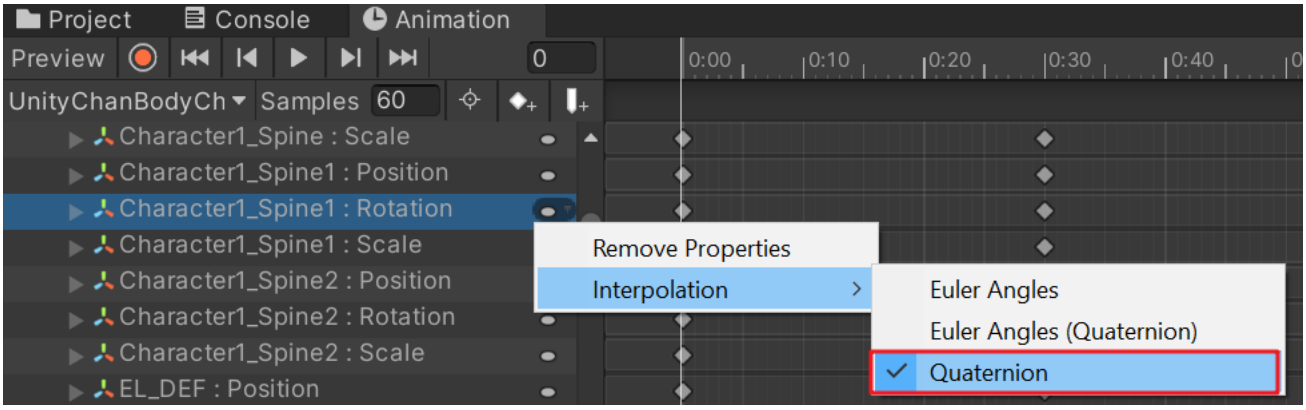
Solution: *ScalableBonesmanager* will update ScalableBones according to their order in the hierarchy, if the order of ScalableBones in the hierarchy is messed up, this problem will occur. Click the "recover" button, then click "Auto Setup" to restore the normal order.

4.3 Unexpected movement of bones

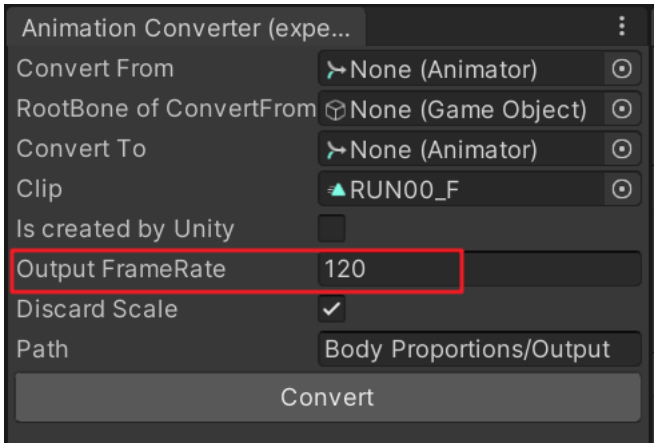
Solution: Please check if there are any other components or shaders affecting the position of the mesh.

4.4 The bone is twisted in opposite orientation.

Solution: If the animation that produces the error was created by you in unity, change the frame interpolation method of the rotation animation to "Quaternion".



If the animation that produces the error was created by Animation Converter, try to increase the output frame rate and convert the animation again. Interpolation errors can occur in animations that rotate too fast, and a higher output frame rate will reduce the possibility of errors.

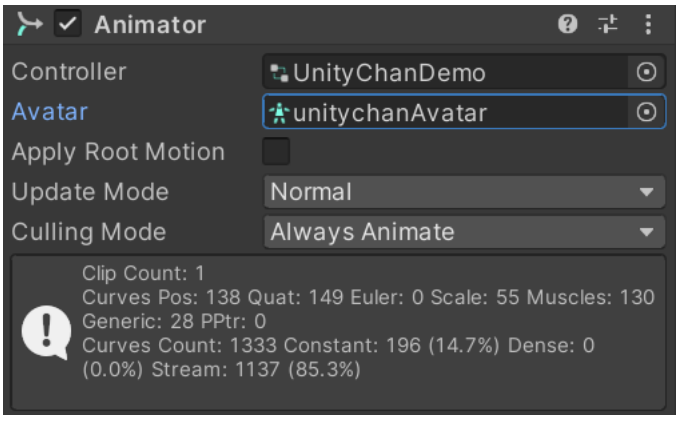


4.5 Animator property missing in a converted animaiton, but animation works normally.

In order to be compatible with both humanoid and non-humanoid animations, as well as to be able to correctly animate the root and hip skeleton at all times, the **AnimationConverter** saves both Animator animation and Transform animation. If the avatar reference is none, the Transform animation will also move the root and hip correctly.

You have two ways to get rid of the annoying misses.

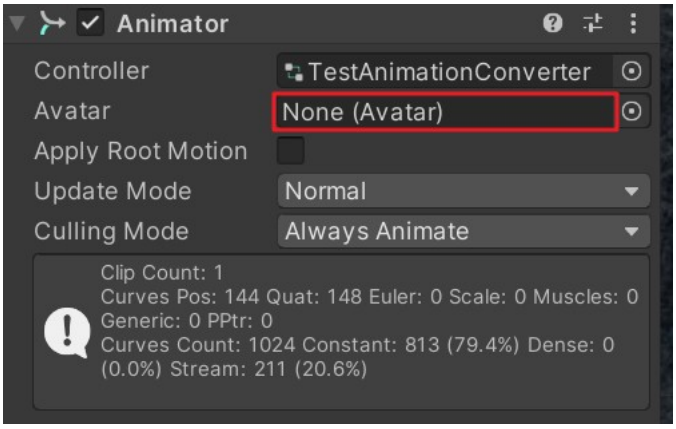
1. Apply the correct avatar. In this case, the Transform animation will not be able to move the roots and hips, but the Animator animation will move them correctly.



2. Leave the avatar empty and remove the missing Animator animation properties.

4.6 Unable to animate root or hip bone of humanoid model

If you want to move root and hip with the Transform animation, Please set avatar to none. Read **4.5** to learn more .



Author: OnlyNew Studio

Website: www.onlynew.tech

email: onlynewstudio@protonmail.com