

## **Dynamic Bone Guide**

### **Wogrim's Brief Guide to Dynamic Bone Physics and the DynamicBone Script.**

#### What is Dynamic Bone?

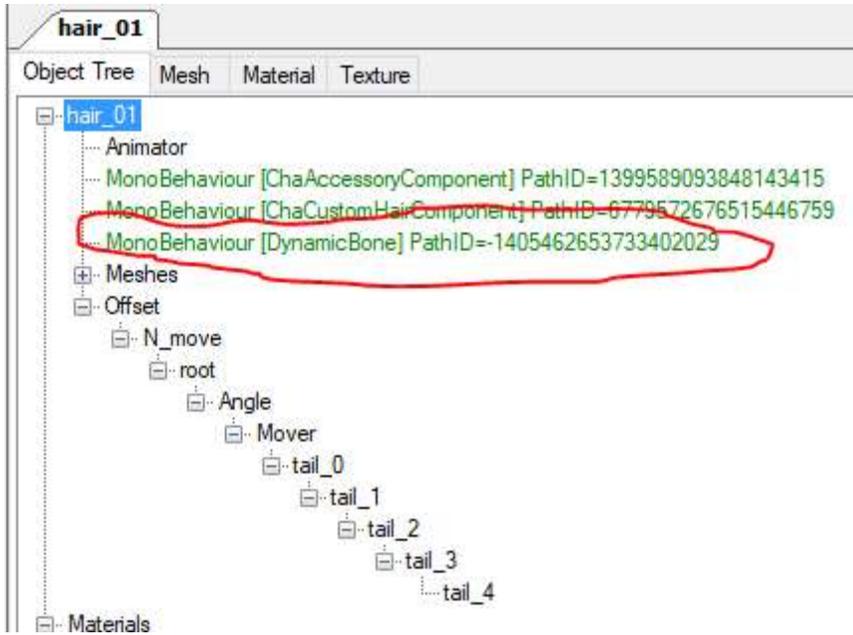
Dynamic Bone is some scripts that game developers can buy on the Unity Asset Store, which lets them put those scripts in their games. The main script is called DynamicBone, which gets applied to a chain of bones such that the bone chain has some physics. It can be something that only jiggles a little like boobs, a springy cat tail, a freely dangling rope, or many other things which you want to move dynamically. This includes collisions with things that have the DynamicBoneCollider script.

These scripts are technically both MonoBehaviours (MBs) but I won't refer to them as MBs to avoid confusion with item MBs (such as ChaClothesComponent = Clothes MB).

## How is Dynamic Bone Used in KK and for modding KK?

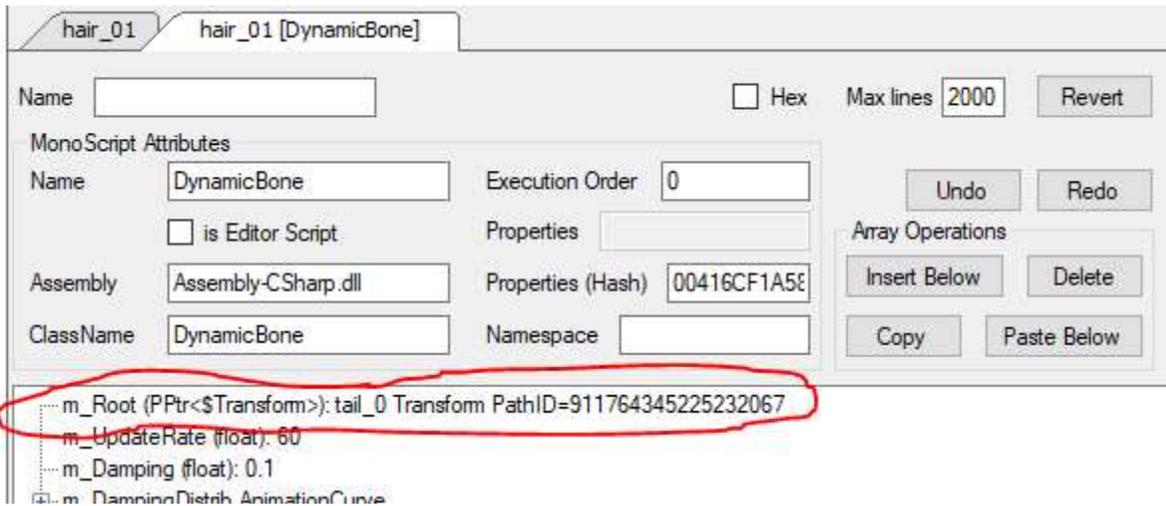
The character body has Dynamic Bone Colliders on it to reduce the amount that Hair, Clothes, Accessories, and Studio Items with Dynamic Bones will clip through the body (at least, the dynamic boned parts of those items). The body also has a DynamicBone\_Ver02 for boobs and butt jiggle; I don't know how it works and this is all I will mention of it.

For making mods you generally put the DynamicBone script on the root of the item (on the same GameObject as the item's MB) so it is easy to find. Here's where you'll find it on an item in SB3U.

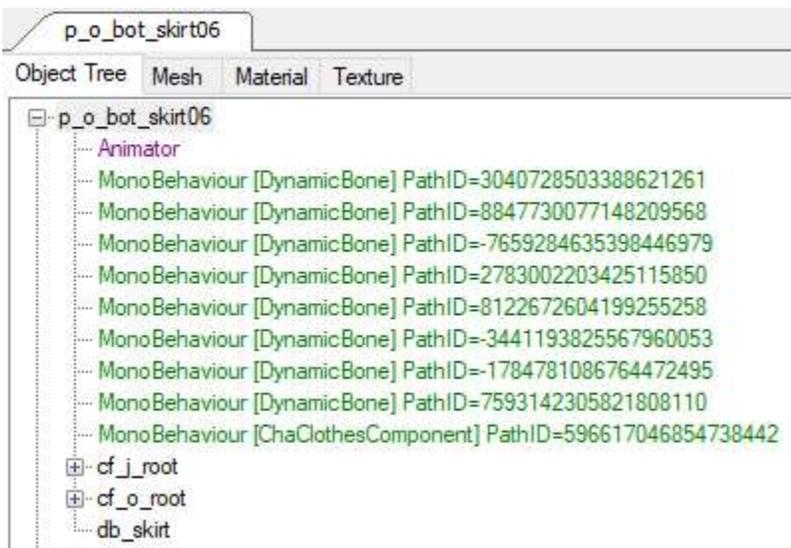


Double click it to look at it.

This one points to the tail\_0 bone so that tail\_0 and its children move with physics.



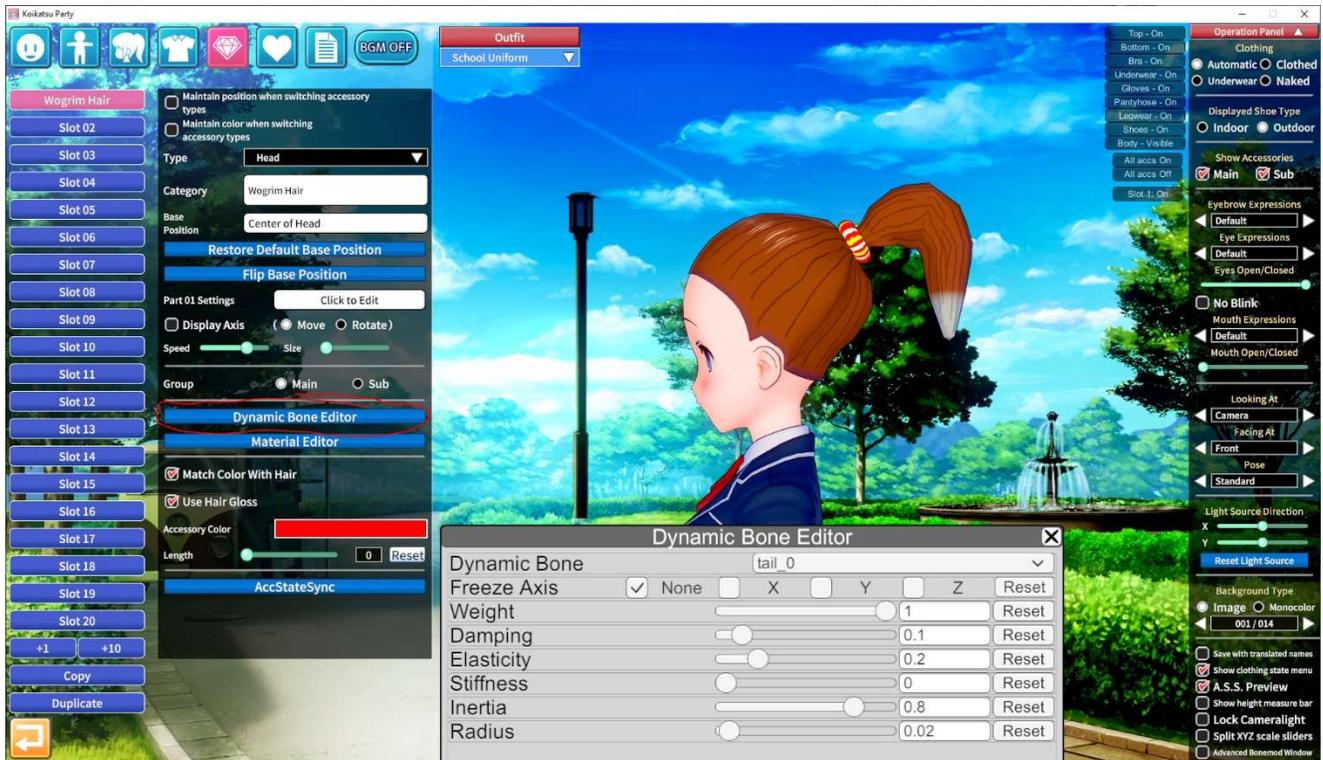
If an item has multiple bone chains that need physics, a separate DynamicBone script is needed for each. For example, skirts that use the body's skirt bones (which are 8 bone chains) usually have 8 DynamicBone scripts on them.



## How To Test Dynamic Bone Settings?

Note: you may be tempted to jump right in to testing, but you would be lacking critical information. I recommend you read the whole guide first, or at least until I've explained all the settings (bold words).

KK Modding Tools doesn't let you test Dynamic Bone settings in the Unity Editor, so you have to test them in-game. For Accessories (including Accessory Hair), you can play with some of the settings in Maker with the Dynamic Bone Editor plugin.



Keyword "some" because this is not all the settings. Also there isn't much character movement in Maker so you don't get a good idea of the result.



## So What Are All the Settings?

Let me start with the less-important things which you probably shouldn't touch:

- **UpdateRate** just leave at 60
- **Colliders** is a list of Dynamic Bone Colliders that is filled by the game at runtime; leave it empty
- **Exclusions** is supposed to let you turn off Dynamic Bone for specific bones in the chain, but I haven't got it to work; leave it empty
- **FreezeAxis** lets you prevent bones from moving in a direction; usually leave it on None
- **DistantDisable / ReferenceObject / DistanceToObject** is a feature that turns off the simulation if it is too far from the reference object (normally used in large game worlds for performance); leave it off / empty
- **notRolls** seems to be a list of bones but I don't know what it's supposed to do; leave it empty
- **Weight** apparently lets you blend between regular animation and the dynamic bone animation, which you could possibly use on a self-animated item; it is not saved in the item and you normally leave it at 1
- **Gravity** puts a constant force (in world space; Y = up/down) on the bones, but there's something weird about how it works and you should probably leave it at 0
- **Force** puts a constant force on the bones, no strings attached; usually used for gravity with something around Y = -0.004 (but people will change it in Studio for posing)

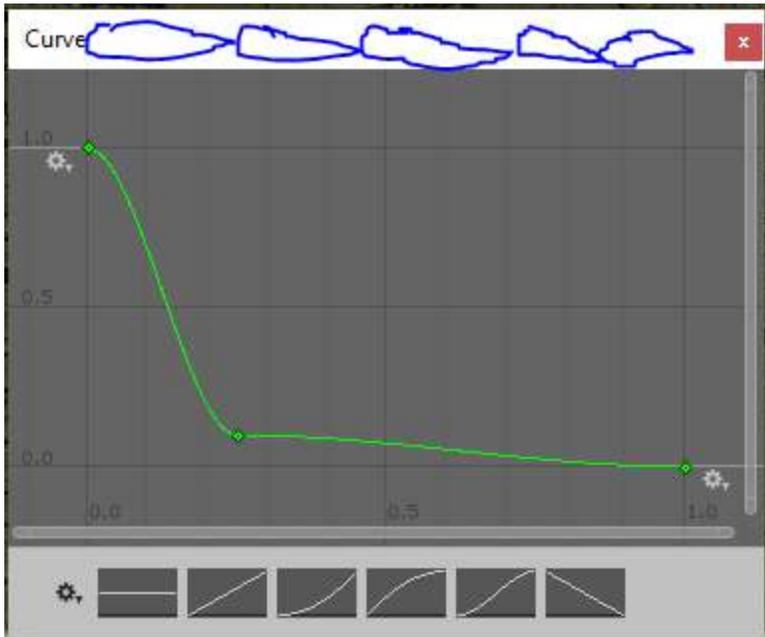
The **Root** is the starting bone in the bone chain that Dynamic Bone will affect; must refer to a bone on your item (you must ALWAYS set this on every Dynamic Bone script)

Now for the main Dynamic Bone settings, and some values to try:

- **Damping** slows the bone movements; 0 for hyper movement, 0.1 is a good starting value, go higher to make something look more lightweight or to make movement less jerky
- **Elasticity** is a force that pushes bones toward their original rotation, and not in the local-rotation way I expected; 0 for no effect, 0.2 is a good amount for the item to mostly defy 0.004 Force (gravity) but still wave with character animation
- **Stiffness** limits how far from the original rotation a bone can rotate, again not in the way I expected; 0 for no limitation, 0.3 appears to be about 90 degree rotation allowed
- **Inert** decreases how much the character movement affects the bones; not animations, but actual movement from one location to another like girls walking around the school; 1 to ignore character movement, 0.8 is a good starting amount, lower to make it look like the wind is blowing it more
- **Radius** is the radius of a sphere collider that gets attached to each bone in the chain, for simulating collisions with the dynamic bone colliders; seems to be 0.01 = 1 cm so try to keep it around half the thickness (diameter) of the item; if too big the bones will collide with the character from far away; I don't know where on each bone the collider is attached

And then there are curves (yes, more settings).

The main Dynamic Bone settings all have distribution curves so that for each setting, different bones in the chain can have different values. These are called **DampingDistrib**, **ElasticityDistrib**, **StiffnessDistrib**, **InertDistrib**, and **RadiusDistrib**. The curves are used on many items; the most common use I've seen is to have high Elasticity at the start of the bone chain (where it attaches to the character) so it doesn't move as much there, but much less Elasticity on the rest of the bone chain so it can move around much more freely. Here's how such a curve could look in Unity (KK Modding Tools), with a 5 bone chain drawn at the top for reference:



The X axis goes from 0 (start of the bone chain) to 1 (end of the bone chain) so basically a bone's X coordinate is where the bone lines up with. Each bone's elasticity is Elasticity multiplied by the Y value of the curve at the bone's X coordinate. So with how much the curves can change things, testing dynamic bones without knowing the curves can cause confusion; for learning it is best to start with no curve (or a flat curve at Y = 1).

**Important notes:**

- from testing, the bone's X coordinate seems to be the tail of the bone; for the first bone in the chain, X = 0.2, second bone 0.4, etc. (these may be based on bone length, I just have all bones same length); so for the above curve, you probably want the high part to extend farther to the right to affect the first bone
- by default Dynamic Bone seems to ignore the last bone in the chain, which I assume is because Dynamic Bone doesn't know how long it is; the last bone does not move, and the bone coordinates are calculated as if there are 4 bones (for first bone, X = 0.25, etc.); you can get the 5th bone working as expected if you set one of these:
  - **EndLength** if non-zero will use the last bone with a length of this number multiplied by the distance between the last 2 bones; use 1 if the last bone is the same length
  - **EndOffset** if non-zero will use the last bone with its tail at the specified offset from it; I haven't tried it but it's probably for making the last bone at a different angle in Dynamic Bone calculations

In SB3U the curve is just some data points with X Value ("time"), Y Value ("value"), and left and right slopes ("inSlope" and "outSlope") for calculating the curve between the data points:

```
m_Elasticity (float): 0.2
m_ElasticityDistrib AnimationCurve
  m_Curve vector
    Array Array size 3
      [0] data Keyframe
        time (float): 0
        value (float): 1
        inSlope (float): 0
        outSlope (float): 0
      [1] data Keyframe
        time (float): 0.25
        value (float): 0.1
        inSlope (float): 0
        outSlope (float): 0
      [2] data Keyframe
        time (float): 1
        value (float): 0
        inSlope (float): 0
        outSlope (float): 0
    m_PreInfinity (int32): 2
    m_PostInfinity (int32): 2
    m_RotationOrder (int32): 0
```

So yeah the SB3U workflow is much more of a pain when dealing with curves.

### Making Items With Dynamic Bones (General)

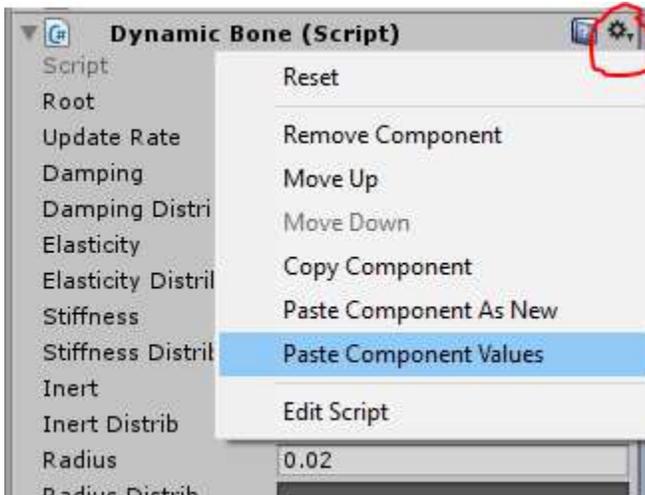
Dynamic Bones can act unpredictable with scaling. Using game controls to scale Accessories and Studio Items (the same amount in all directions) is probably okay, but you don't want any scaling anywhere on your item when you create it.

Dynamic Bones may be put on a bone chain that branches out and it seems to work, but I don't know how that affects calculations; do so at your own risk.

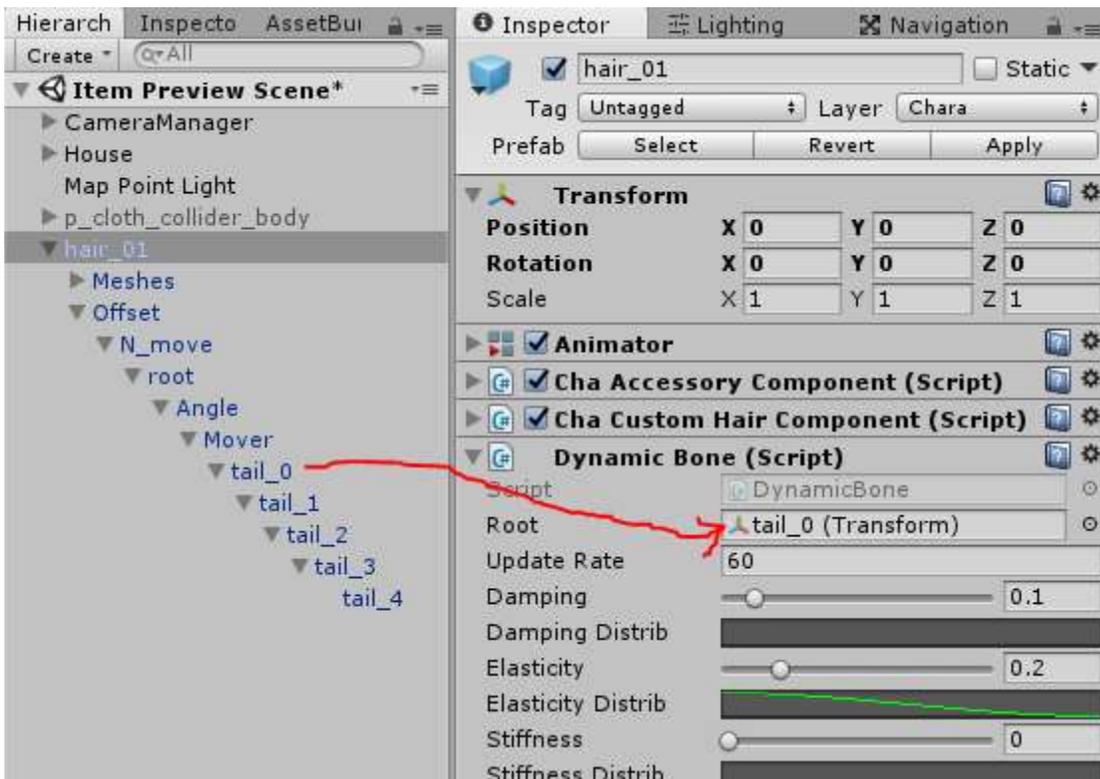
I've heard you're not supposed to attach something with dynamic bones to the dynamic bones of something else, something with unknown script execution order; do so at your own risk.

## Making Items With Dynamic Bones (KK Modding Tools)

The Dynamic Bone script can be added directly to your item in KK Modding Tools but can't be tested until in-game. For editing curves, right-click is your friend. Once you have settings you like on an item, you can copy/paste them on any new items you make:



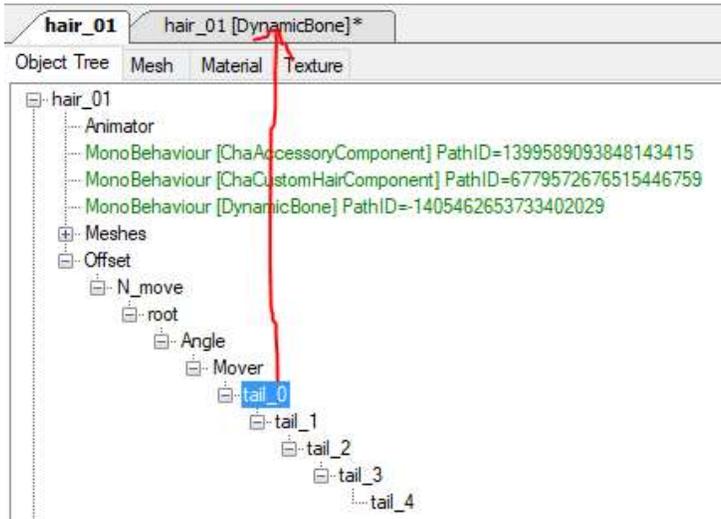
You still have to drag the appropriate bone to the Root field after pasting settings:



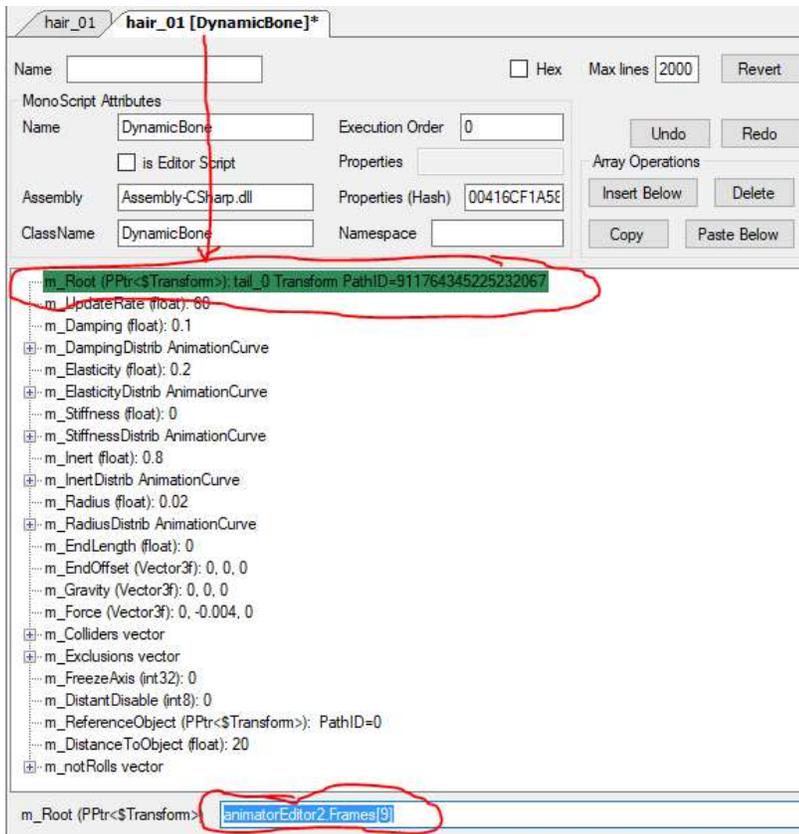
If for some reason you want the exact settings of a game item or mod which you only have the AB for, go to the KK Modding Tools GitHub, Wiki section, "Importing Contents Of AssetBundles" page.

## Making Items With Dynamic Bones (SB3U)

As with other scripts in SB3U, to add the Dynamic Bone script to an Animator you must drag it from an existing item that has the script. This copies the settings, which makes things easy if there is an existing item whose settings you want to copy. But to fill in the Root field you must drag the bone from the Animator's Object Tree to the Dynamic Bone script's tab...



And onto m\_Root, which will fill in the link at the bottom; hit Enter to change the value, and it will turn green.



If you need to change how many data points are in a curve, you can probably do it with Array Operations, but I haven't tested the resulting item. I imagine the points need to be in order of increasing X value ("time").

hair\_01 hair\_01 [DynamicBone]\*

Name   Hex Max lines 2000 Revert

MonoScript Attributes

Name DynamicBone Execution Order 0 Undo Redo

is Editor Script Properties

Assembly Assembly-CSharp.dll Properties (Hash) 00416CF1A5E Array Operations

ClassName DynamicBone Namespace  Insert Below Delete

Copy Paste Below

m\_Root (PPtr<\$Transform>): tail\_0 Transform PathID=911764345225232067

- m\_UpdateRate (float): 60
- m\_Damping (float): 0.1
- m\_DampingDistrib AnimationCurve
- m\_Elasticity (float): 0.2
- m\_ElasticityDistrib AnimationCurve
- m\_Stiffness (float): 0
- m\_StiffnessDistrib AnimationCurve
- m\_Curve-vector
  - Array Array size 3
- m\_PreInfinity (int32): 2
- m\_PostInfinity (int32): 2
- m\_RotationOrder (int32): 0