

# OpenVRInputDocs

[www.vreue4.com](http://www.vreue4.com)

<b>1. Pre-Setup .....</b>	<b>3</b>
<b>2. Main Setup .....</b>	<b>3</b>
<b>3. Input Mapping .....</b>	<b>5</b>
<b>4. Gesture Detection .....</b>	<b>5</b>

## 1. Pre-Setup

1.

If you are running the module WITHOUT VRExpansionPlugin and are not using the branch specifically intended for this, then open up OpenInputPlugin.build.cs and change the line

```
bool bCompileWithVRExpansion = true;
```

to

```
bool bCompileWithVRExpansion = false;
```

This will cause it to no longer compile against my primary plugin and to act standalone.

2.

Alter your DefaultEngine.ini file to include this:

```
[SystemSettings]
vr.SteamVR.EnableVRInput=0
vr.SteamVR.EnableSteamVRInputController=1
```

This turns off beta input and enables the custom beta input of the new SteamVRInputController shipped with this module.

The skeletal action names used are below (auto set if the skeletal action name is blank):

```
/actions/main/in/righthand_skeleton
/actions/main/in/lefthand_skeleton
```

I am setting the plugin to default to these for now when none have been provided by the end user.

You can access the local binding interface from your desktop here:

<http://localhost:8998/dashboard/controllerbinding.html> or by following the link in this website to it (added for convenience).

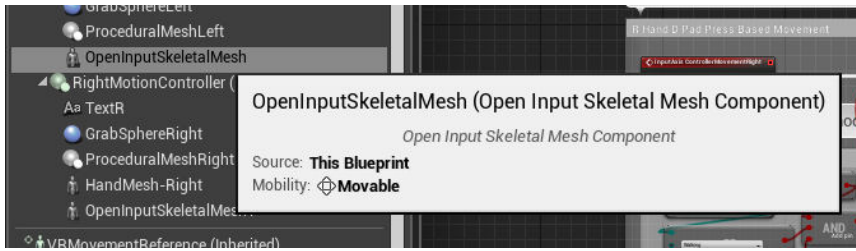
## 2. Main Setup

You may want to get the knuckles controller profile from the latest template (if using the VRExpansionPlugin).

Also I am also packaging the OpenInput testing hands as they are pre-rigged and fully setup already and they look 100x better than pretty much all stock hands for ue4. (when using ue4 hands you likely will want to turn off bone morphing in the settings as they likely won't be rigged correctly for it).

1.

Load in and go to your pawn, under each motion controller add an OpenInputSkeletalMeshComponent



Add at least 1 skeletal action to the skeletal action array and set its hand type (Left or Right under SkeletalData)

If you have a custom action manifest generation then add the action names in, if you do not then they will be auto filled with my defaults.

Leave the rest of the settings default for now.

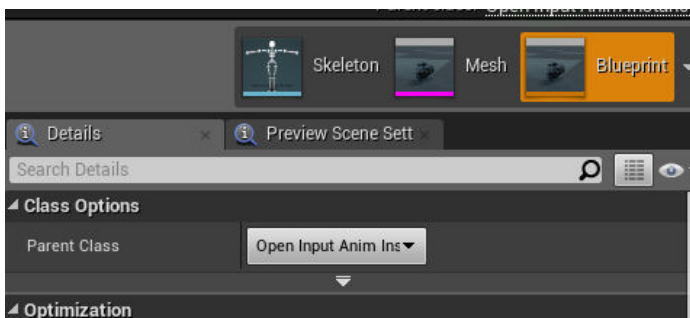
2.

Set the mesh and animation blueprint for the mesh

Open up the animation blueprint for your hand meshes

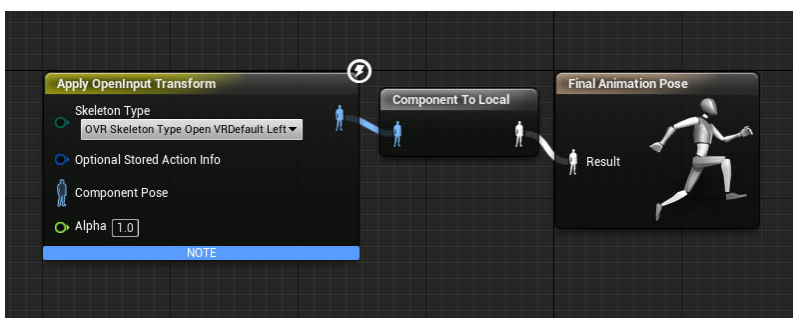
Click on “ClassSettings” on the menu bar

Re-parent the class to the OpenInputAnimInstance class



3.

Go to the Anim graph for the anim blueprint and drop in a Apply\_OpenInput\_Transform animation node



Hook up the output to the final animation pose input

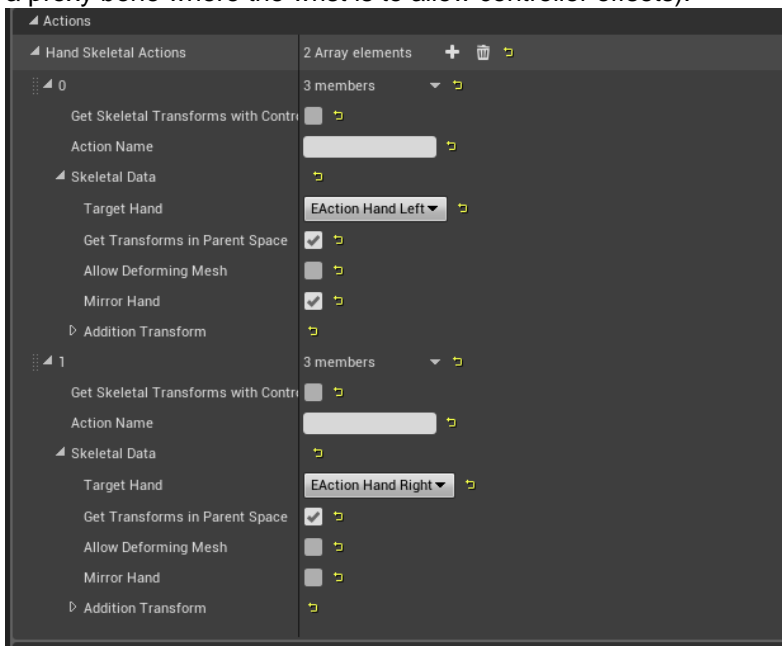
Change the drop down selection to which hand right/left it is (I may automate the hand type at some point) and which skeleton to use OpenInput/UE4

You can also manually map a new skeleton but that is for advanced use.

4. Full body setup

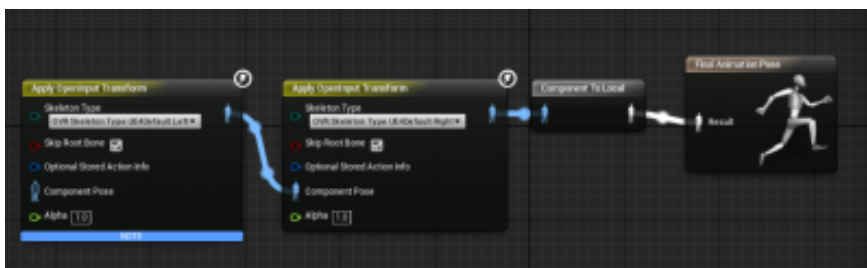
If you are doing a full body setup you will need two skeletal actions and two Apply\_OpenInputTransform nodes in the animation graph, one for the left and one for the right hands.

You will also likely want to tick on bSkipRootBone on these nodes so that the wrist isn't moved (unless you have a proxy bone where the wrist is to allow controller offsets).



Also for the left hand you will likely need to tick the "MirrorSkeletal" boolean on the action struct as the OpenInput skeletal data is not setup for inverse directions.

You also will want GetTransformsInParentSpace true and AllowDeformingMesh false (if the hand mesh isn't weighted and rigged correctly).



### 3. Input Mapping

Coming soon

### 4. Gesture Detection

Coming Soon