

Dynamic Asset Selection

Use Choosers Tables and Proxy Assets to dynamically select Assets such as Animations at runtime.

 Learn to use this **Experimental** feature, but use caution when shipping with it.

You can use a combination of **Proxy**, **Proxy Table** and **Chooser Table** assets to build dynamic asset selection logic, to drive animations based on variables in your project. For example, you can use Proxy and Proxy Table assets to select what kind of animation set should be loaded and used for your character, such as different weapon sets. Alternatively, you can use Chooser Tables to dynamically select individual animation assets, such as different hit reactions based on the location your character is hit, using contextual variables from your project.

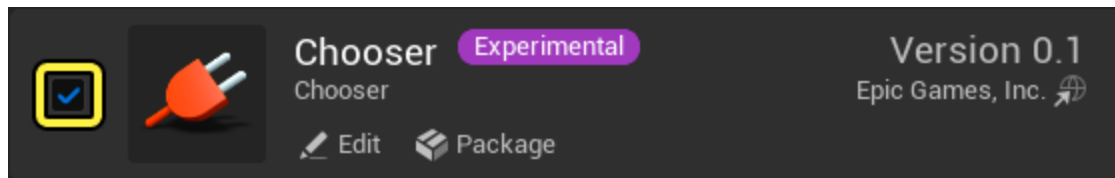
You can use the following document to learn more about setting up a dynamic animation selection system in Unreal Engine.



Though this documentation focusses on using Choosers and Proxy Tables to select animation related assets such as Animation Sequences, Montages, or AnimInstance classes, the system itself is generic and can be used to select any type of Asset, Object, or Class.

Prerequisites

- Enable the **Chooser** plugin. Navigate in the **Menu Bar** to **Edit > Plugins** and locate **Chooser** in the **Animation** section, or use the **Search Bar**. Enable the plugin and restart the editor.



- Your project contains a set of animations you wish to dynamically select based on contextual situation at runtime. This could be unique cosmetic animation sets, relevant gameplay animations such as contextual animation sineros, or animation sets relevant to equipable items like weapons.
- Your project contains a functional Animation Blueprint, that you can build dynamic Animation selection logic in.

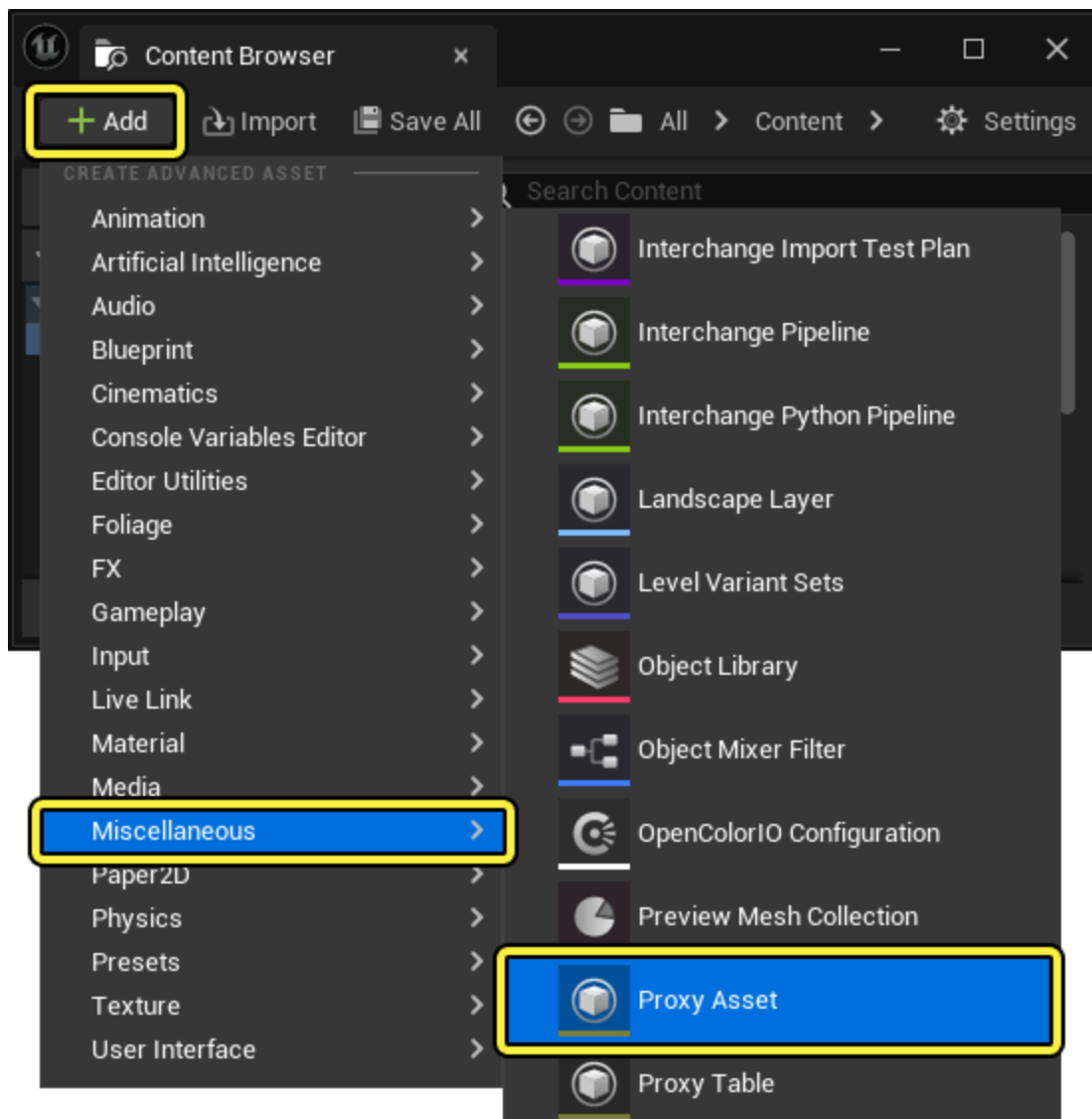
Set Up an Animation Selection System

Here you can read about how to set up an animation chooser system in your project to dynamically select hit reaction animation playback based on context in your project at runtime.

Create a Proxy Asset

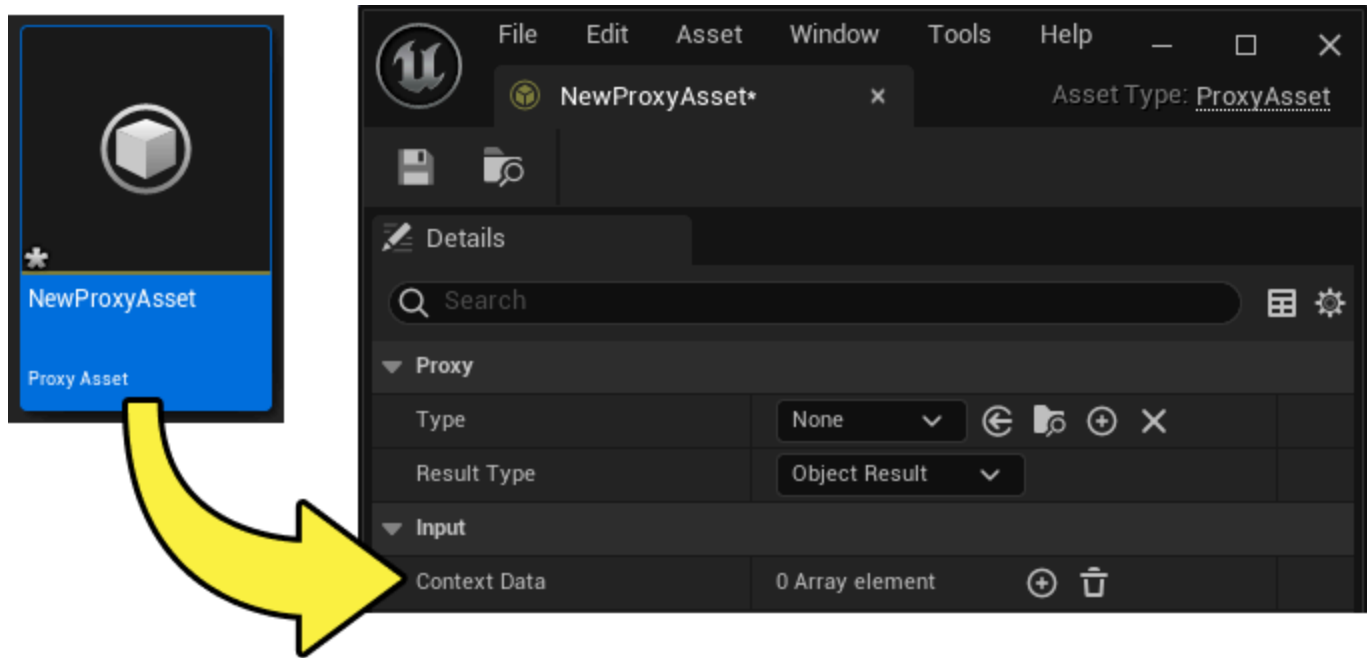
A [Proxy Asset](#) is used to store the contextual information of which [Proxy Table](#) asset is active, and other relevant variables.

To create a Proxy Asset, use (+) **Add** in the **Content Browser** and navigate to **Miscellaneous > Proxy Asset**.

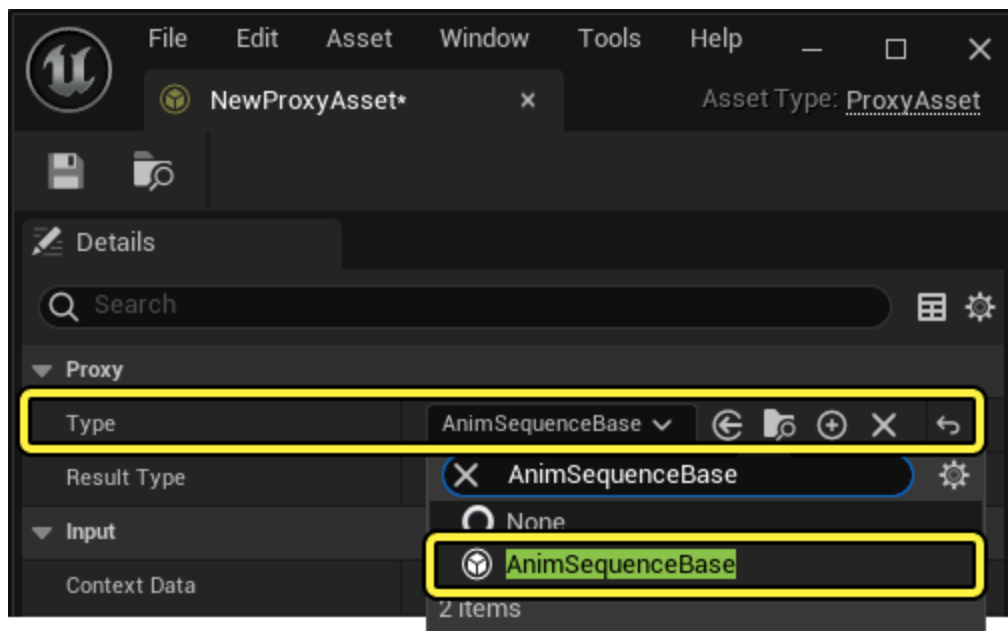


Create a Proxy Asset for each animation set you wish to dynamically select animations from to drive the character, such as an idle, walk, or run set.

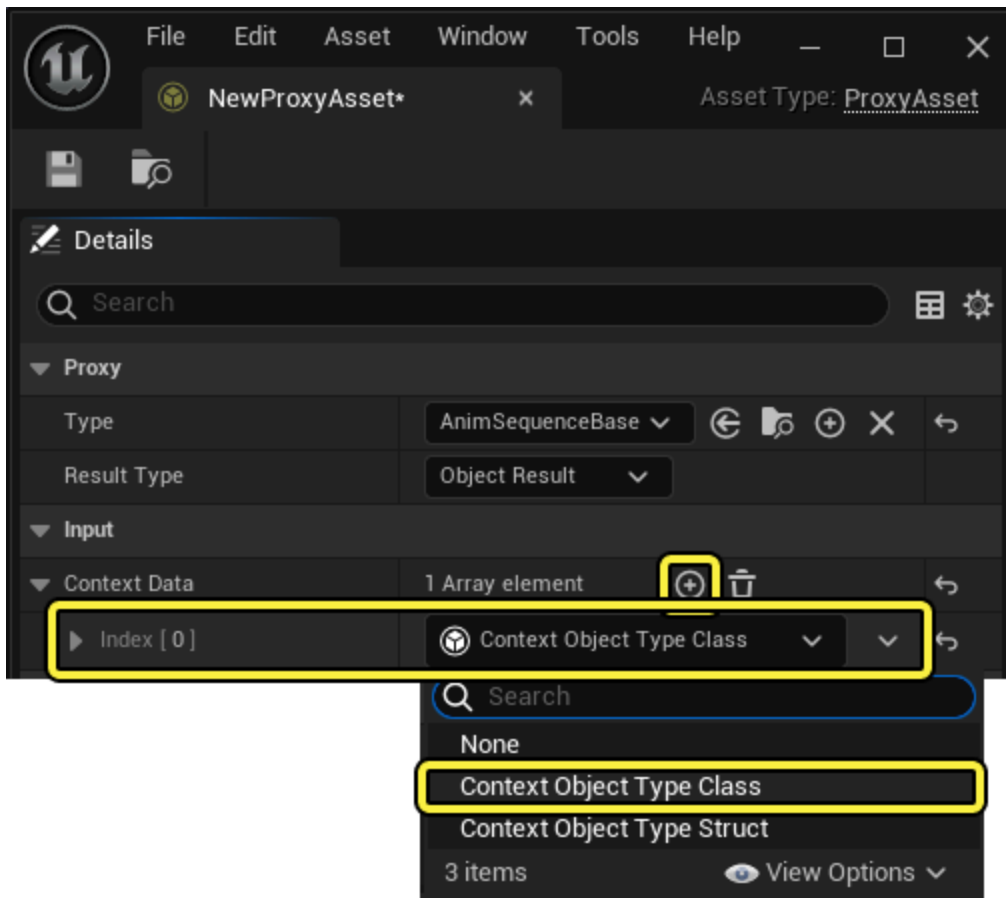
After creating the Proxy Assets, open each to access their settings.



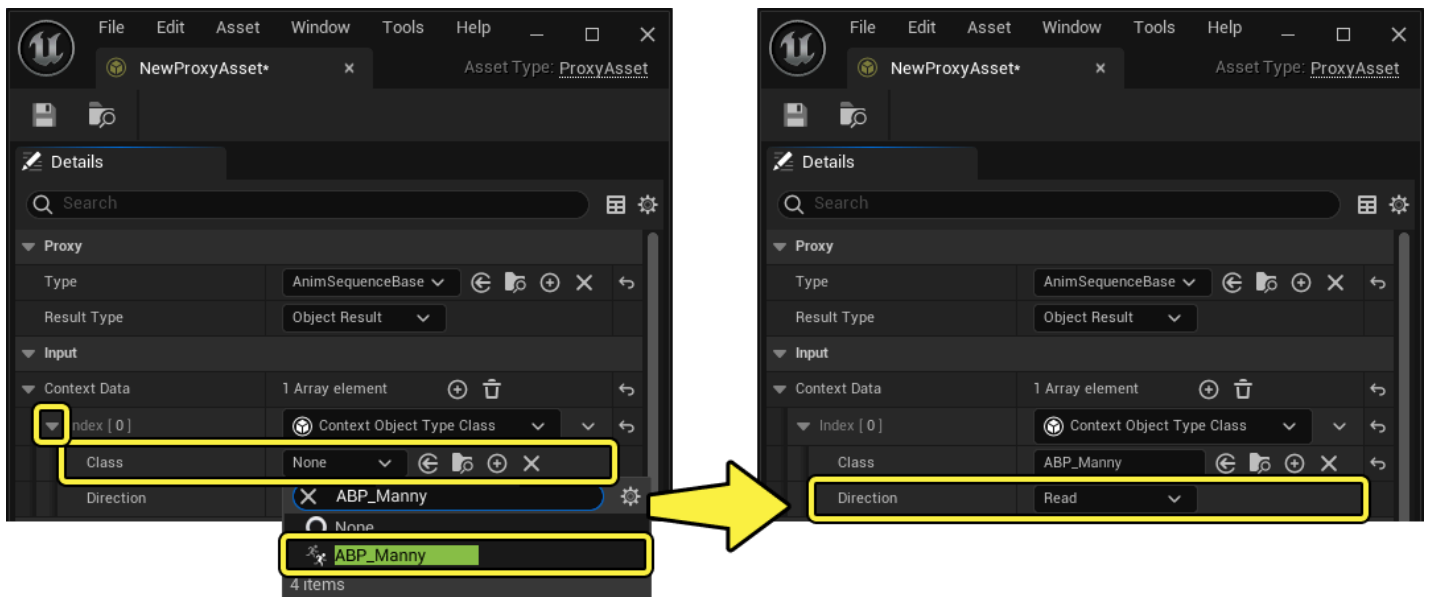
For each Proxy Asset, set the **Type** property to the animation asset type you are using, in this case `AnimSequenceBase`.



Set the **Context Data** property to the **Animation Blueprint** option by adding a new **Index** array using (+) **Add**, selecting **Context Object Type Class** option.



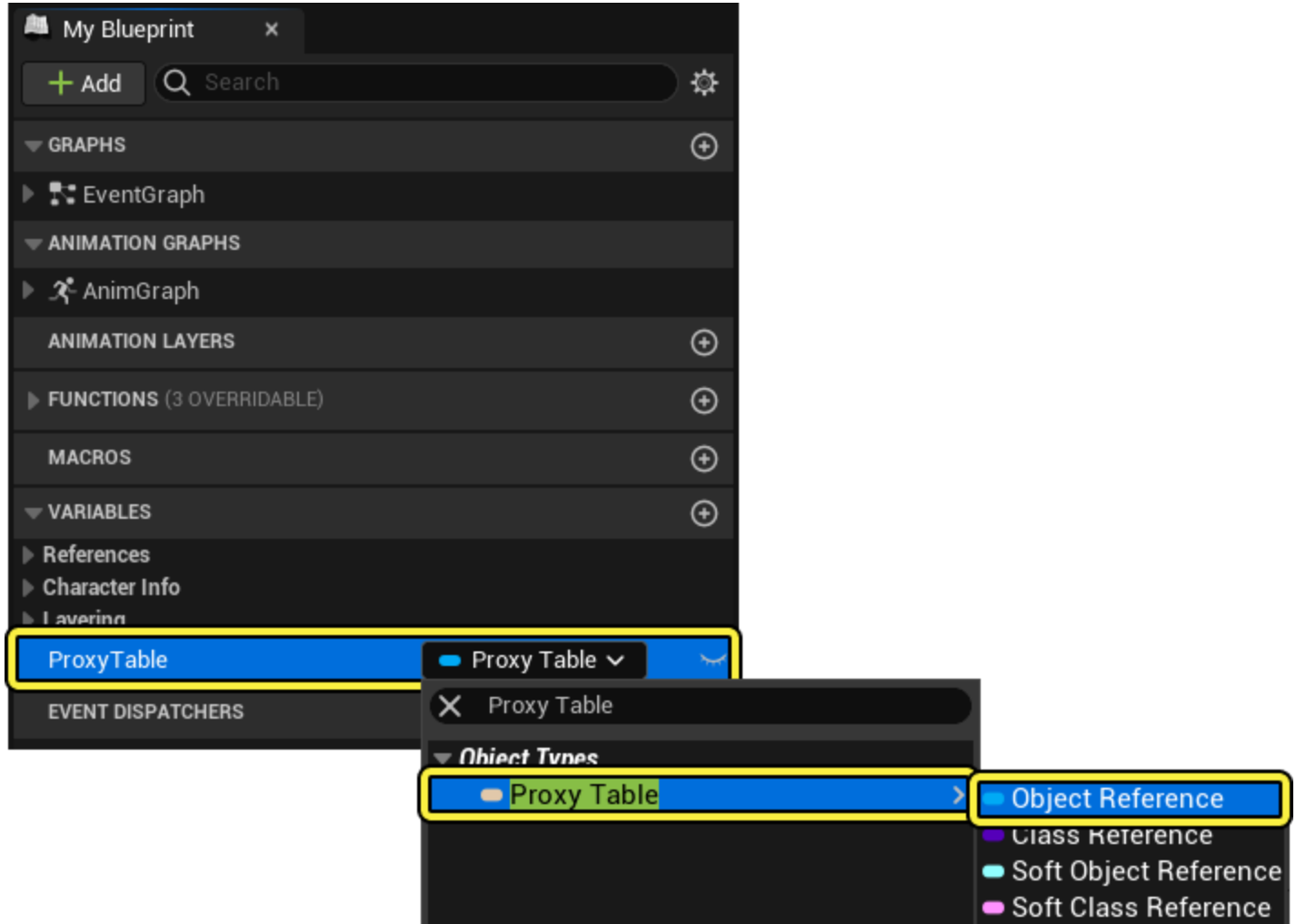
Expand the **Index** array's settings and set the **Class** property to use your Animation Blueprint, in this case `ABP_Manny`. Then ensure the **Direction** property is set to **Read**.



Animation Blueprint Set Up

After setting up your Proxy Asset, you must create a variable in your character's animation blueprint in order to store the active proxy table during runtime. To create this variable, open

your character's Animation Blueprint and create a new Variable in the **My Blueprint** panel using (+) **Add**. Then set the Variable type to a **Proxy Table Object Reference**. After creating the variable, **save** and **compile** your Animation Blueprint.

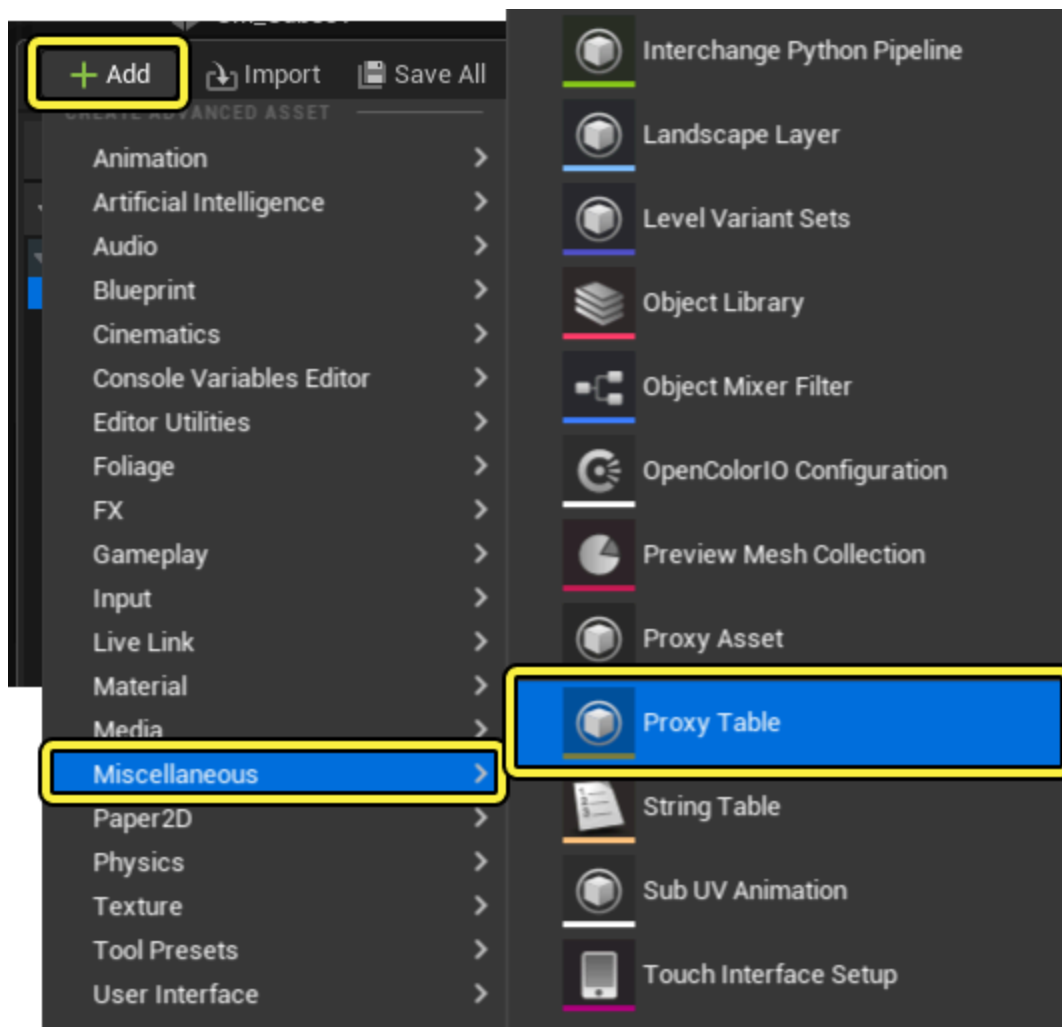


The Proxy Table variable will be used in your animation graph by an Evaluate Proxy node, to determine the active Proxy Table at runtime.

Create Proxy Table Assets

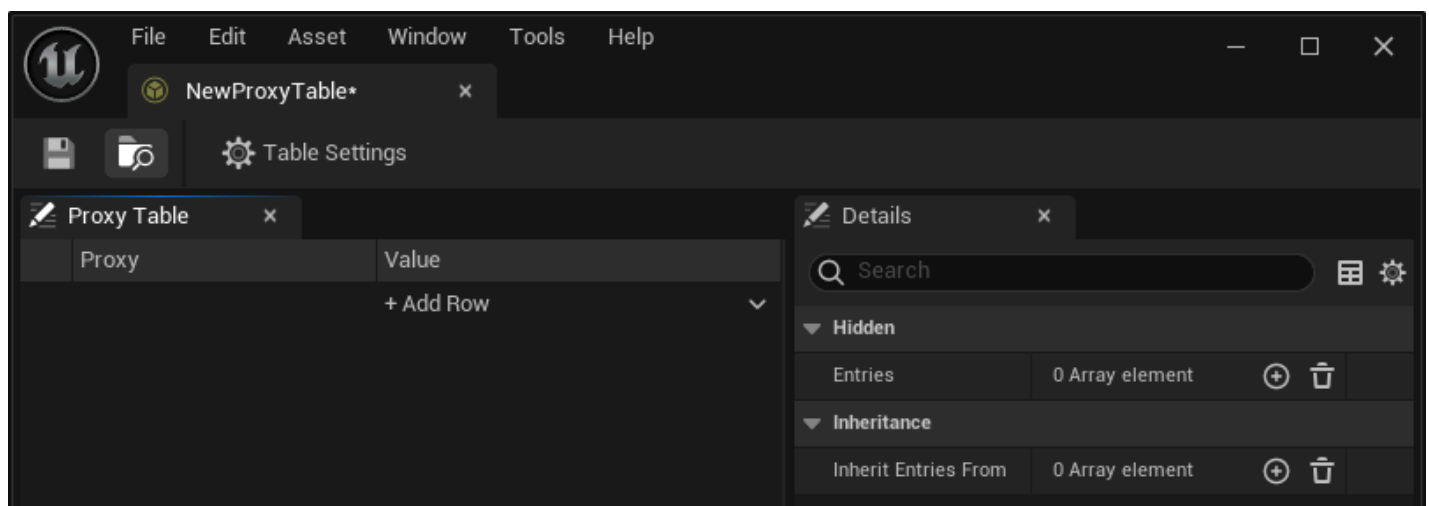
A **Proxy Table** asset is used to store the animation assets sets, that can be dynamically selected at runtime. For example, one Proxy Table may store a character's Idle animations while another may store its Walking or Running animation set.

To create a Proxy Table asset, use (+) **Add** in the **Content Browser** and navigate to **Miscellaneous > Proxy Table**.

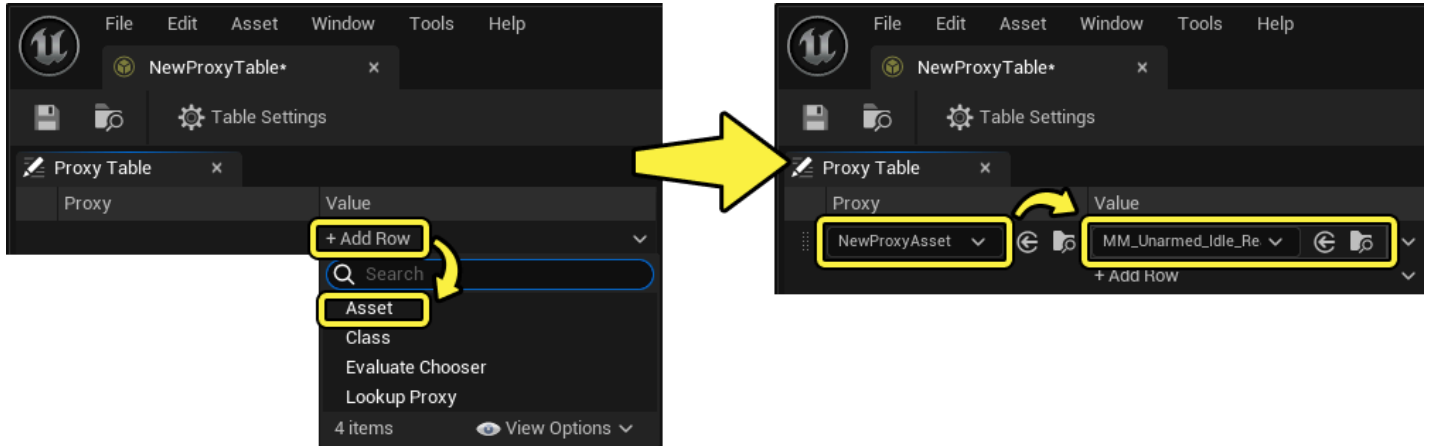


Create a Proxy Table asset for every contextual situation that a different animation set would be needed, such as locomotion animation sets for unarmed characters and characters holding a pistol or rifle.

After creating a Proxy Table asset, open the asset to access the Proxy Table's values.



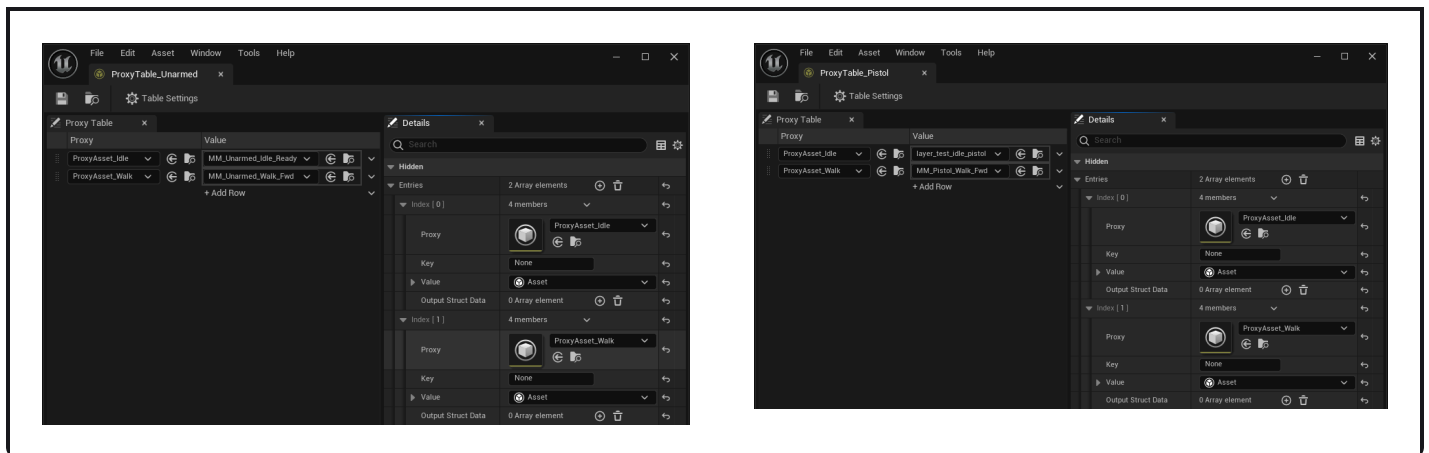
In the ProxyTable, add an entry for each **Proxy Asset** using (+) **Add Row** and selecting asset. You can then assign the associated animation asset from the set using the Value Column.



In the following workflow example, a `ProxyTable_Unarmed` asset is assigned an `Unarmed_Idle` and an `Unarmed_Walk` animations to the respective `ProxyAsset_Idle` and `ProxyAsset_Walk` rows. Then in a `ProxyTable_Pistol` asset, the same Proxy Asset rows are assigned with a `Pistol_Idle` and `Pistol_Walk` animations respectively.

ProxyTable_Unarmed

ProxyTable_Pistol

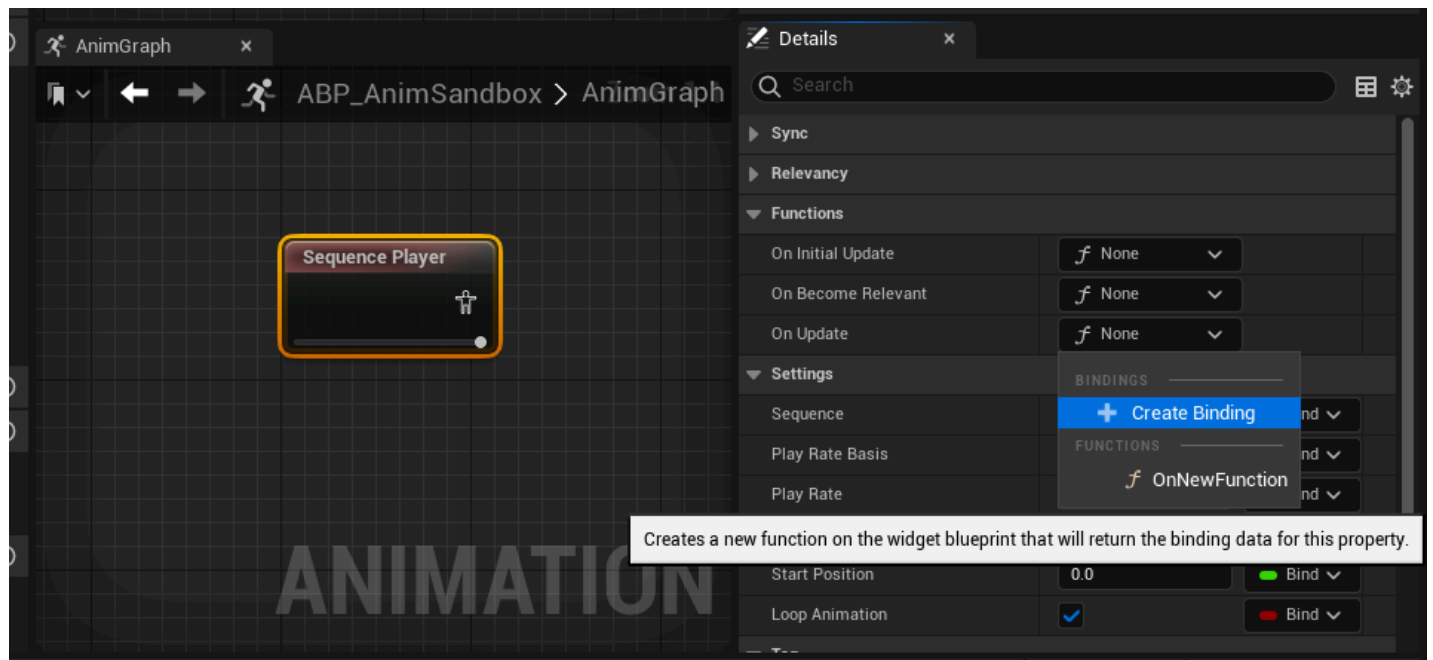


The Value column can also contain a **Class** reference, a **Chooser Asset**, or a **Lookup Proxy** for more dynamic animation selection systems.

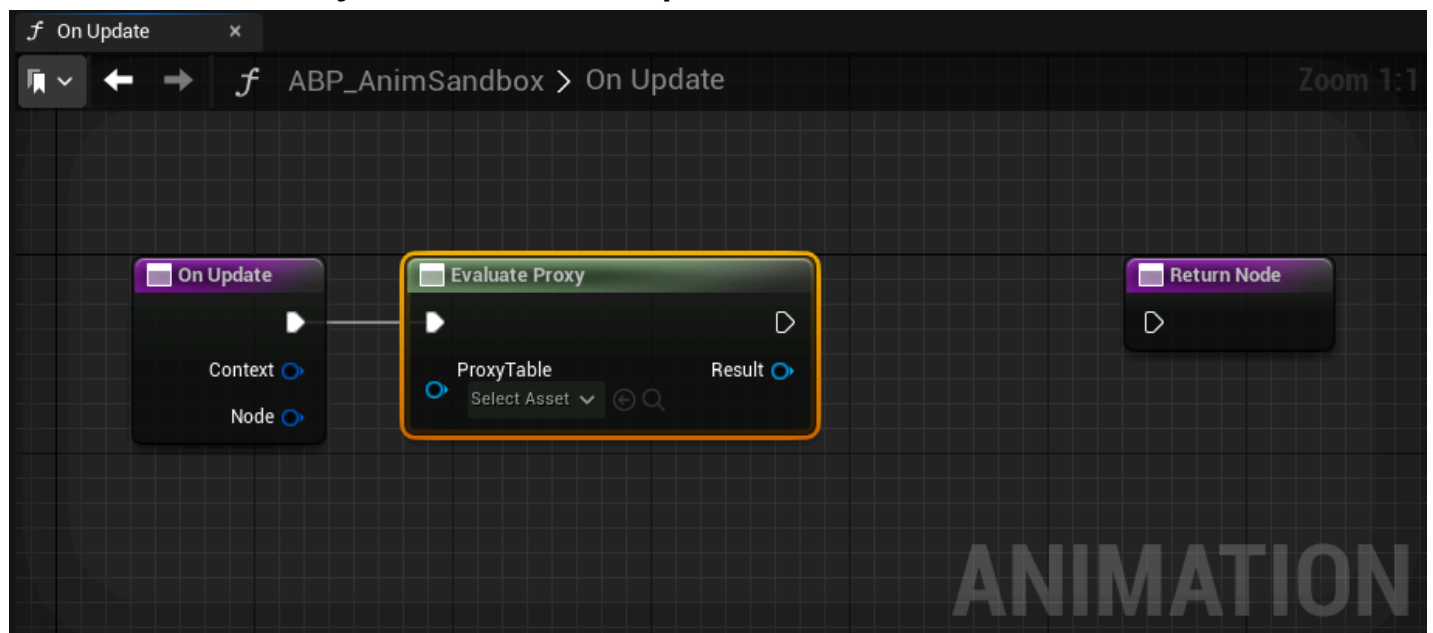
Using Proxy Tables at Runtime

You can use Proxy Table assets at runtime using an [Animation Node function](#) on a Sequence Player node. To create a new function to select a Proxy Table asset, select the Sequence

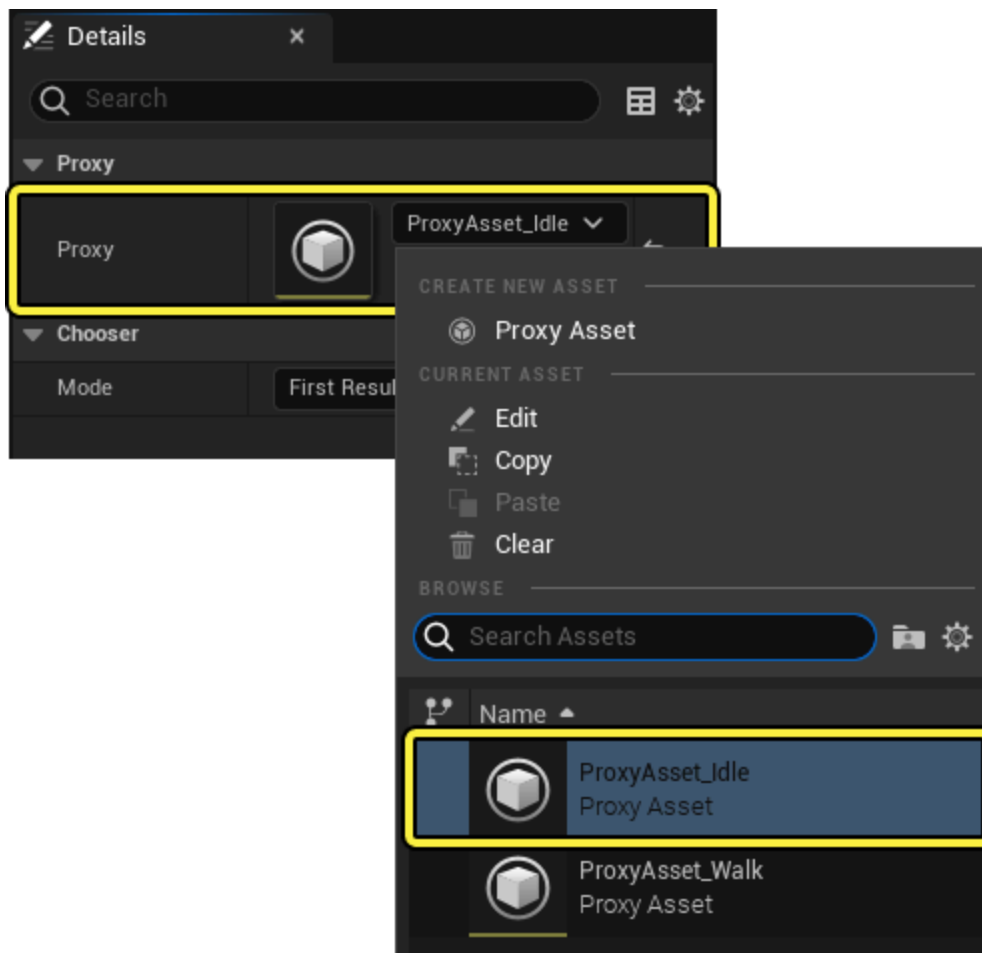
Player node, and add a new Function at the **On Update** binding.



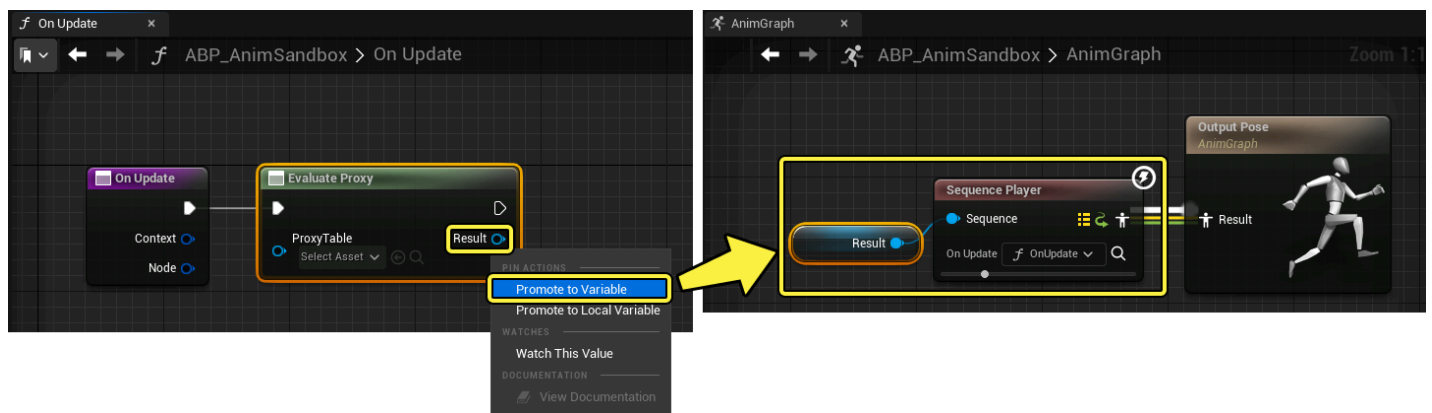
Add an **Evaluate Proxy** node from the **On Update** node.



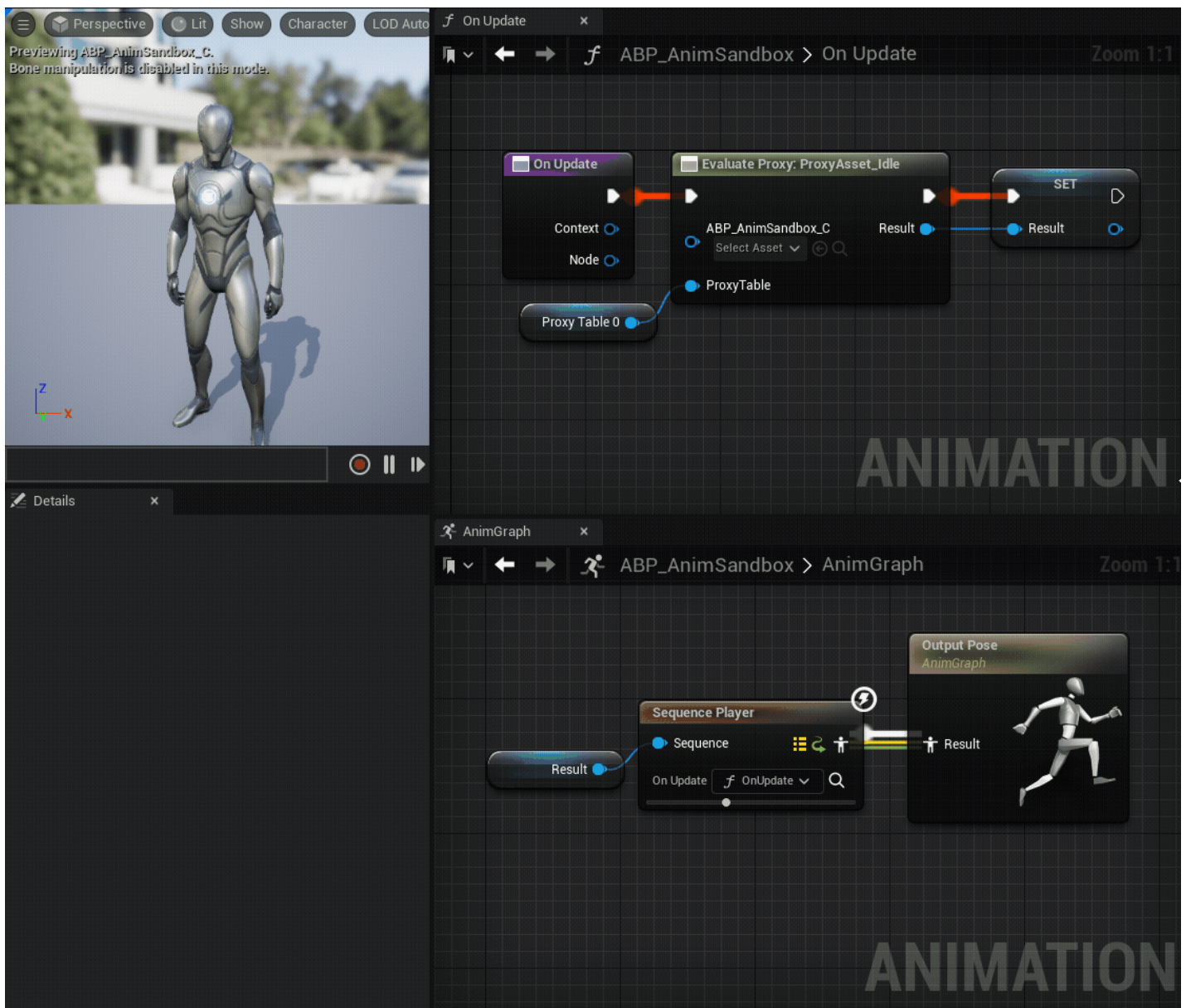
Then select the node and select the Proxy Asset in the Proxy property using the drop down.



Then promote the result to a Variable, and connect the output to the Sequence Player node.



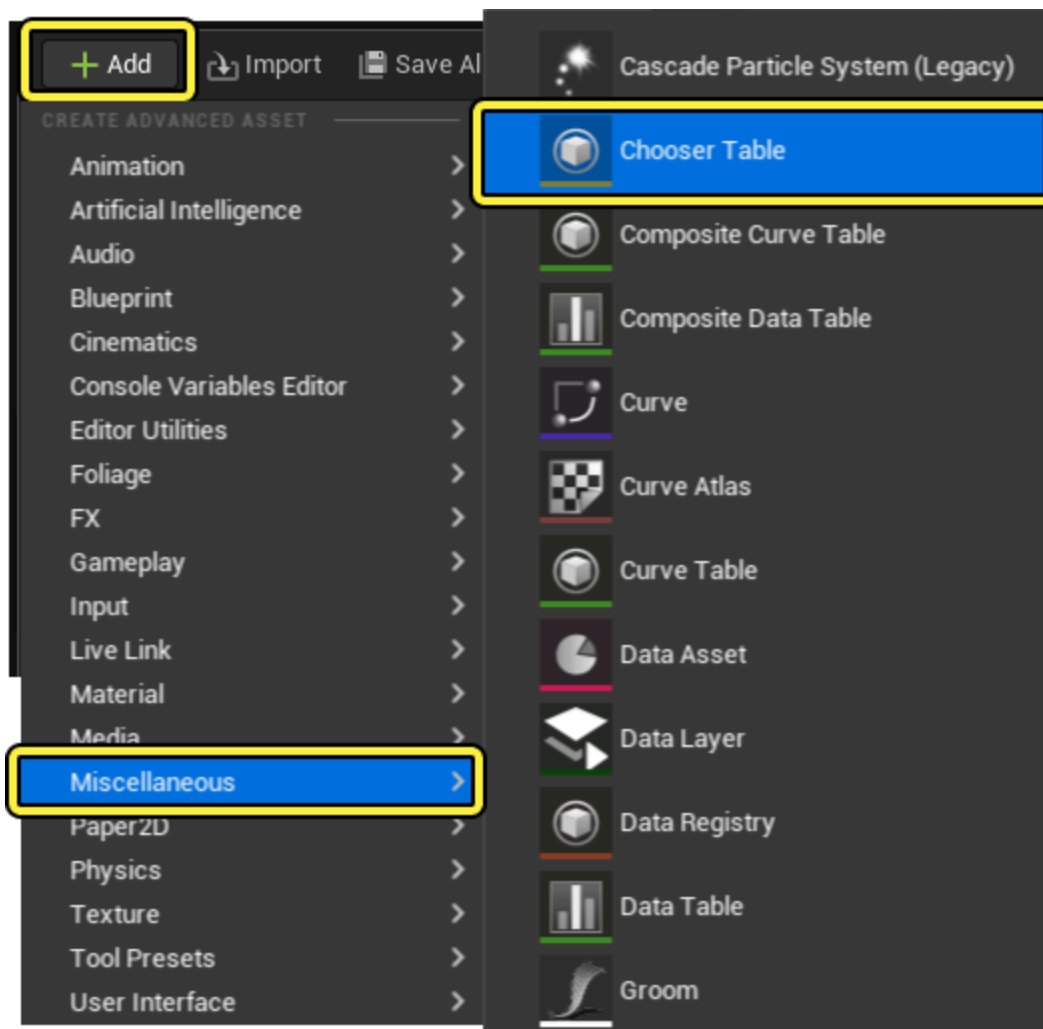
You can then set the active Proxy Table asset on the Evaluate Proxy node using a variety of methods, such as a Chooser Table, to dynamically change the animation set driving the character.



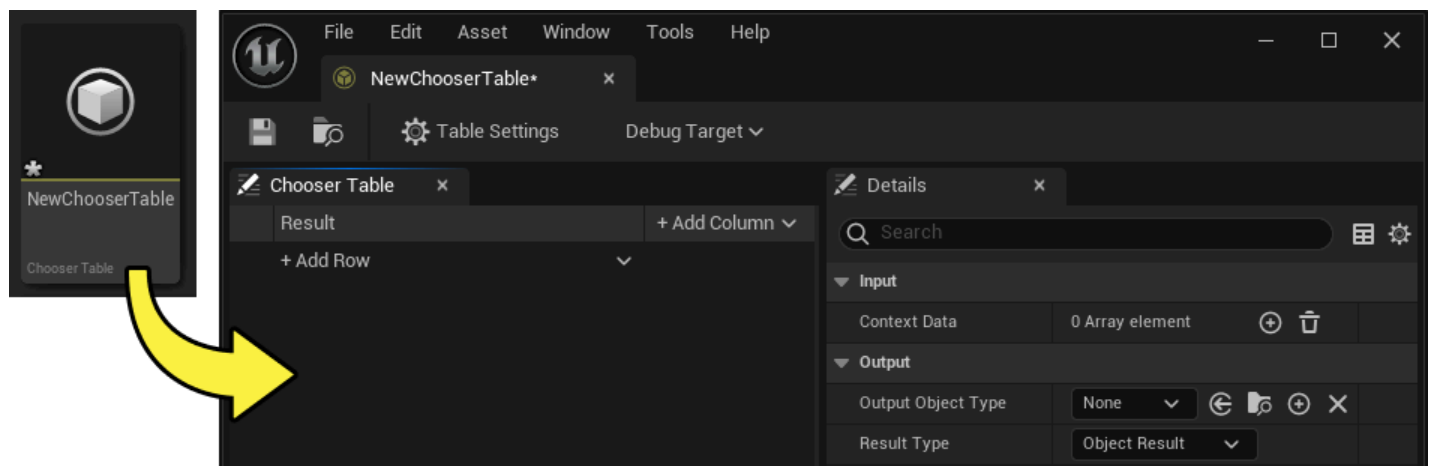
Create a Chooser Table Asset

A chooser asset is used to store animation data sets that comprise of an animations various iterations, that can be selected and played back based on the context. For example, a chooser table may contain a set of hit reaction animations, with each entry being an animation that reacts based on a different part of the body being hit (arms, legs, chest, head), that can then be selected based on a contextual variable, such as the location of the hit.

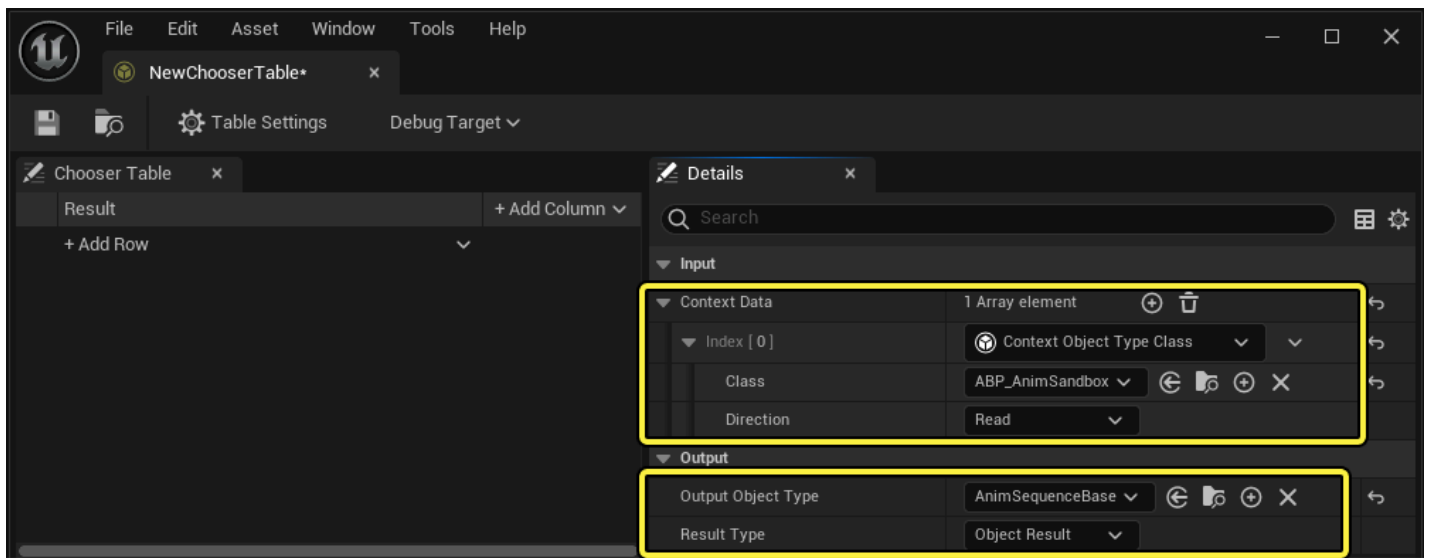
To create a Chooser Table asset, use (+) **Add** in the **Content Browser** and navigate to **Miscellaneous > Chooser Table**.



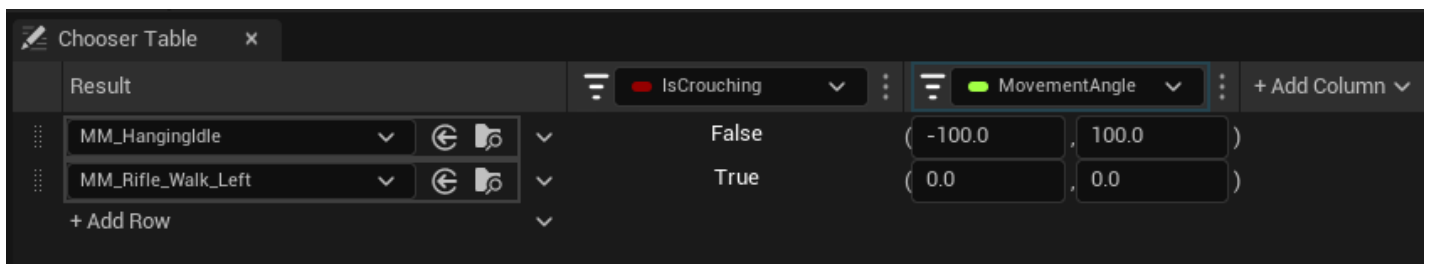
After creating a Chooser Table, you can open the asset to access its properties.



Create a new Array element in the Context Data property using (+) **Add** and set the property to the **Context Object Type Class** option. Then expand the Index array, and set the **Class** property to your character's animation blueprint and ensure the **Direction** property is set to **Read**. You can then set the Output Object Type to the animation asset you are using. This workflow example is using Animation Sequences, so the **AnimSequenceBase** class option was selected.

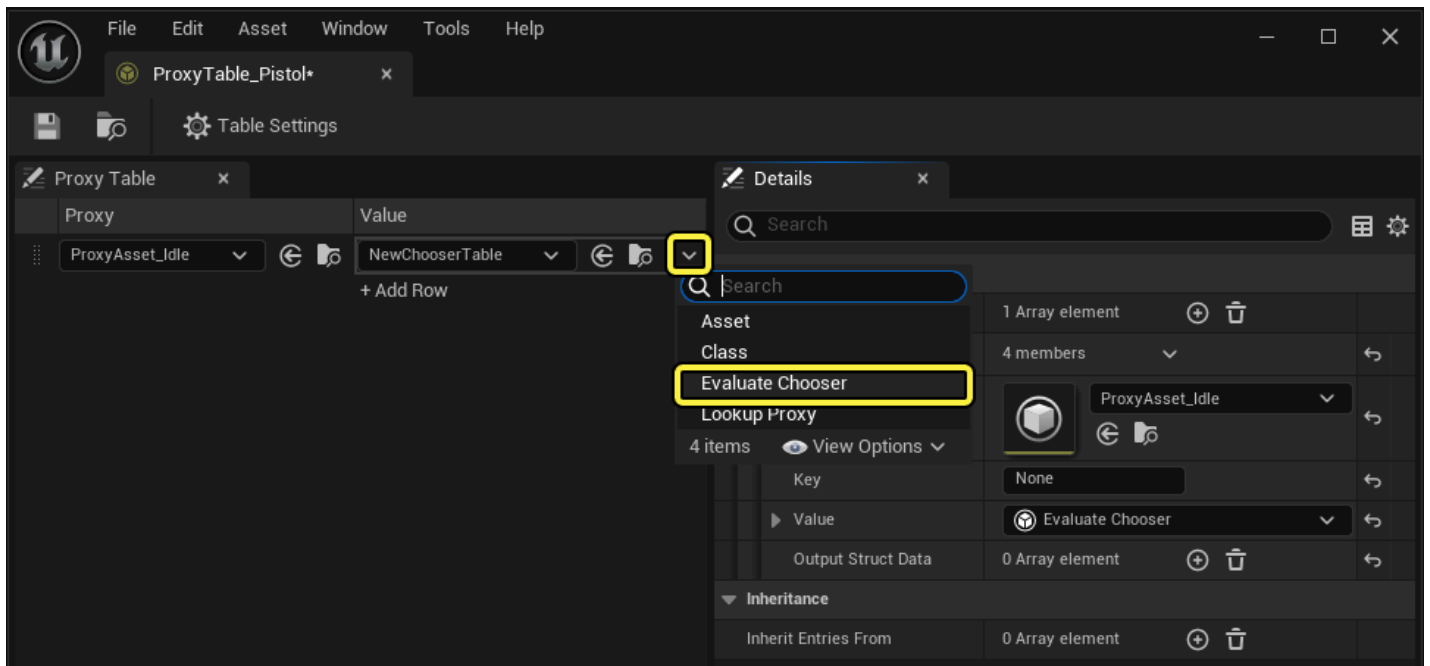


In the Chooser Table panel, you can begin to add Columns, in order to set the Variables from the animation blueprint you wish to influence the selection process using the (+) **Add Column** button. After creating a Column, you can define which Variable from your Animation Blueprint can influence the selection, and the values or states of the variable that must be reached in order to select the Animation Sequence asset within each row.



In this workflow example, the boolean variable `IsCrouching` will select the `MM_HangingIdle` animation when the value is false, and will select the `MM_Rifle_Walk_Left` when the value is true. The `MoveemntAngle` Variable will select the `MM_HangingIdle` when the value is between `-100` and `100`, while the `MM_Rilfe_Walk_Left` animation will only be selected when the value is `0.0`.

In order to drive the characters animations using this selection process, you must set your Proxy Table Entry to **Evaluate Chooser** and assign the Chooser Table asset.



Now the selected animation will change based both on the active Proxy Table asset and the selection made by the Chooser Table.

You can also use ChooserTables without ProxyTables using an Evaluate Chooser node in your animation blueprint graph or state machine.

