Mirroring Animation

Mirror animation in Unreal Engine using the Mirror Data Table.

Animation mirroring copies animation from one side of a character to another so you can reuse the same animation in different situations. Using the **Mirror Data Table**, and other mirroring workflows, you can mirror not only your Animation Sequences, but also curves, sync markers, and Notifies. Additionally, mirroring within Unreal Engine provides a way to create mirrored animations without needing to manage a second copy.

This document provides an overview of how to mirror animation using the Mirror Data Table and Animation Blueprints.

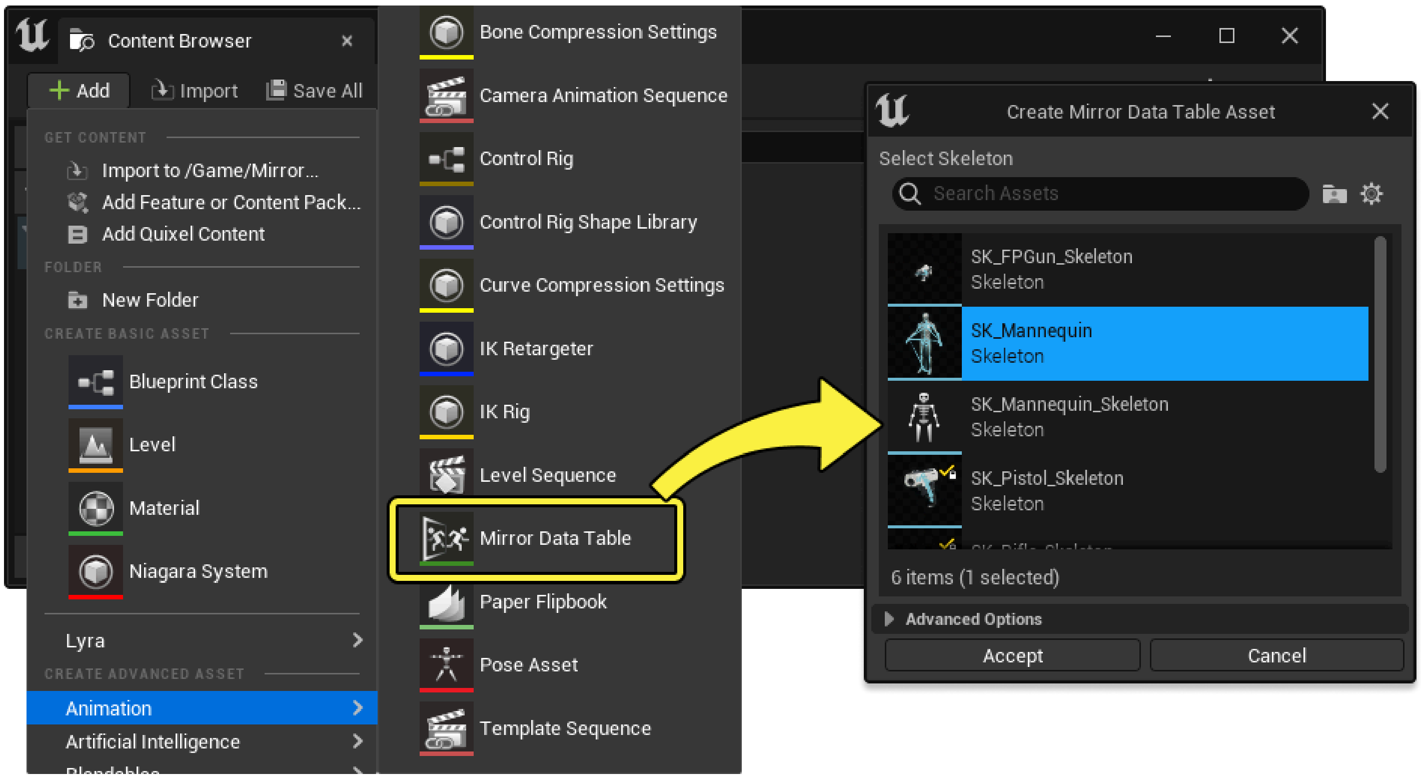
Prerequisites

- Your project already contains a Skeletal Mesh and animations to mirror.
- You have an understanding of how to create and use Animation Blueprints.

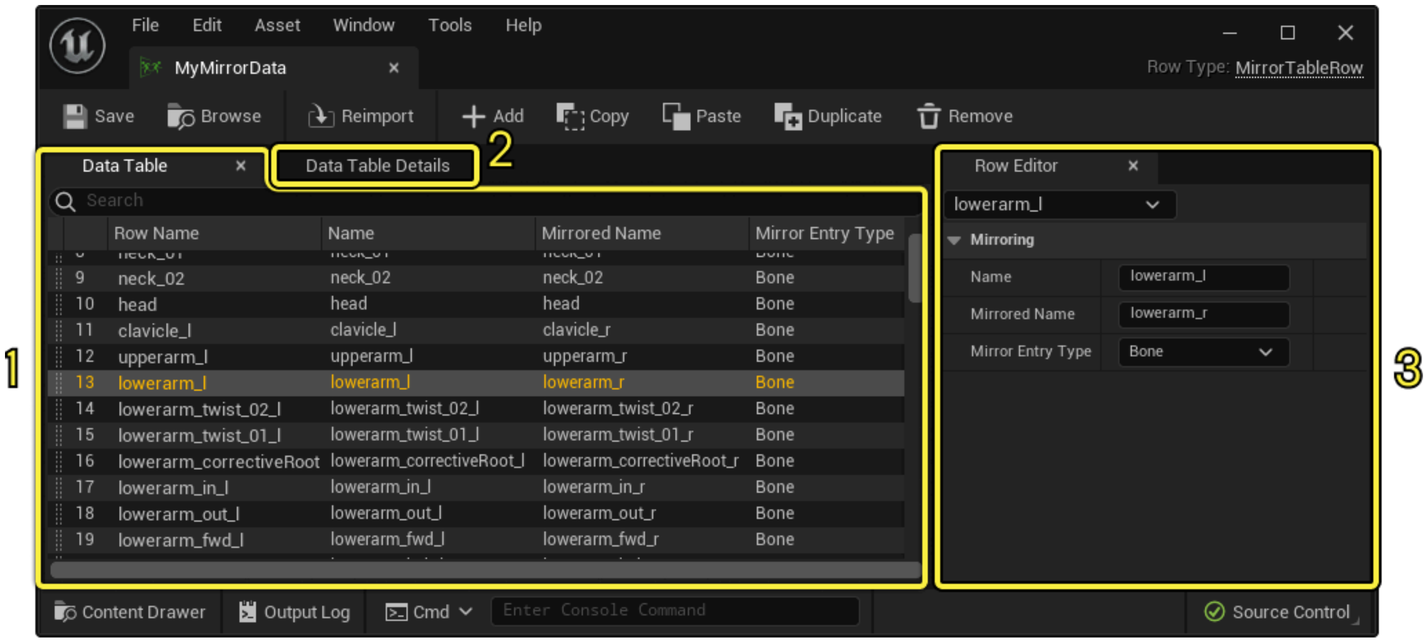
Mirror Data Table

To start mirroring animations, you must first create a **Mirror Data Table** Asset. Mirror Data Tables provide mirroring assignments and instructions for all the elements of a Skeleton you want to mirror.

To create it, click **Add (+)** in the Content Browser and select **Animation > Mirror Data Table**. A dialog window will appear where you must select the Skeleton you want to mirror. Select one and click **Accept**.



Open the Mirror Data Table to view the editor with the following primary areas:



	Row Name	Name	Mirrored Name	Mirror Entry Type
9	neck_02	neck_02	neck_02	Bone
10	head	head	head	Bone
11	clavicle_l	clavicle_l	clavicle_r	Bone
12	upperarm_l	upperarm_l	upperarm_r	Bone
13	lowerarm_l	lowerarm_l	lowerarm_r	Bone
14	lowerarm_twist_02_l	lowerarm_twist_02_l	lowerarm_twist_02_r	Bone
15	lowerarm_twist_01_l	lowerarm_twist_01_l	lowerarm_twist_01_r	Bone
16	lowerarm_correctiveRoot	lowerarm_correctiveRoot_l	lowerarm_correctiveRoot_r	Bone
17	lowerarm_in_l	lowerarm_in_l	lowerarm_in_r	Bone
18	lowerarm_out_l	lowerarm_out_l	lowerarm_out_r	Bone
19	lowerarm_fwd_l	lowerarm_fwd_l	lowerarm_fwd_r	Bone

lowerarm_l

Mirroring

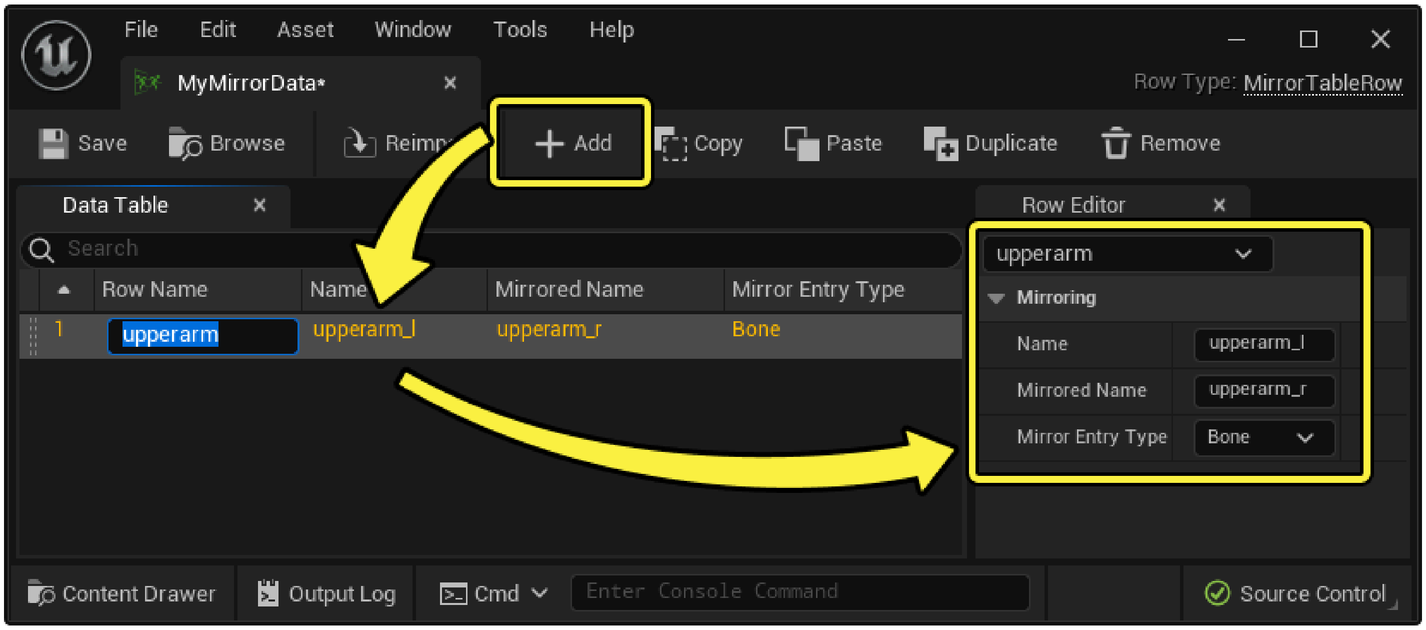
Name	lowerarm_l
Mirrored Name	lowerarm_r
Mirror Entry Type	Bone

1. **Data Table**, which contains the list of elements to mirror. This list auto-populates depending on the bone, notify, and other element names, which you can configure in the Data Table Details.
2. **Data Table Details**, which contains configuration settings for the mirroring behavior.
3. **Row Editor**, which provides settings for the selected entry where you can edit the element name, the mirrored name, and the element type.

Adding and Editing Entries

To add a new table entry, click the **Add (+)** toolbar button and fill the four properties in the **Row Editor** panel.

- **Row Name**, which is the name of the entry.
- **Name**, which is the name of the first bone to mirror.
- **Mirrored Name**, which is the name of the second bone to mirror. Left or right bones can go in either of these properties, but these properties must contain the symmetrical bones.
- **Mirror Entry Type**, which is the element type to mirror. You can select the following types:
  - Bone
  - Animation Notify
  - Curve
  - Sync Marker
  - Custom, which provides a code foundation for extending the Mirror Data Table in C++ by adding additional types.

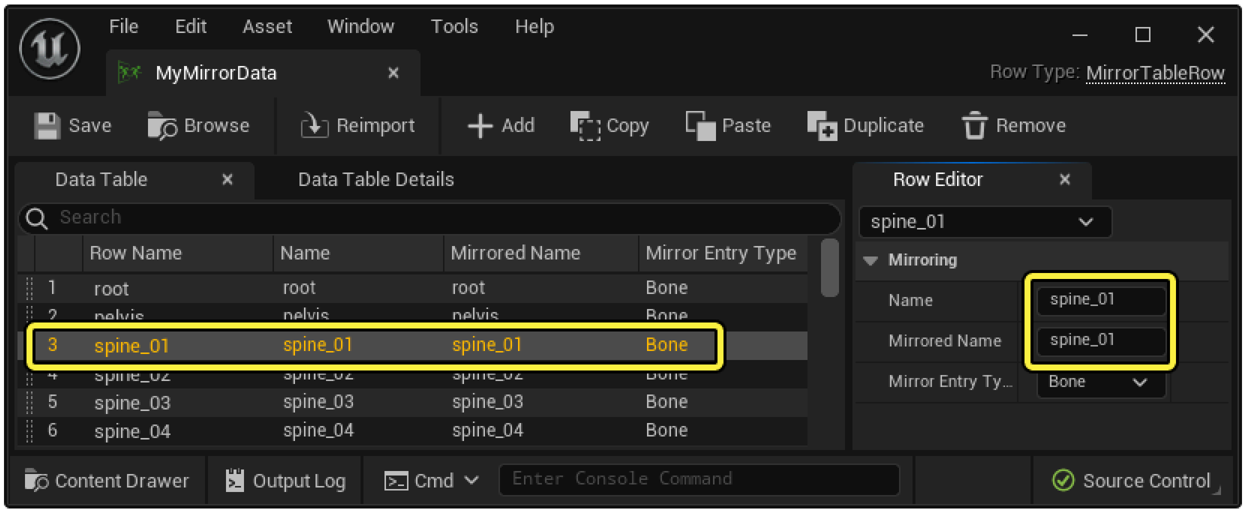


	Row Name	Name	Mirrored Name	Mirror Entry Type
1	upperarm	upperarm_l	upperarm_r	Bone

upperarm

Name	upperarm_l
Mirrored Name	upperarm_r
Mirror Entry Type	Bone

In order to have a fully mirrored character, it is necessary that the table contains most skinned bones, including central bones like **pelvis**, **spine**, **neck**, and **head**. This is so that the mirroring operation can appropriately flip the rotations of those bones along the mirror axis. For these entries, both the **Name** and **Mirrored Name** should match.



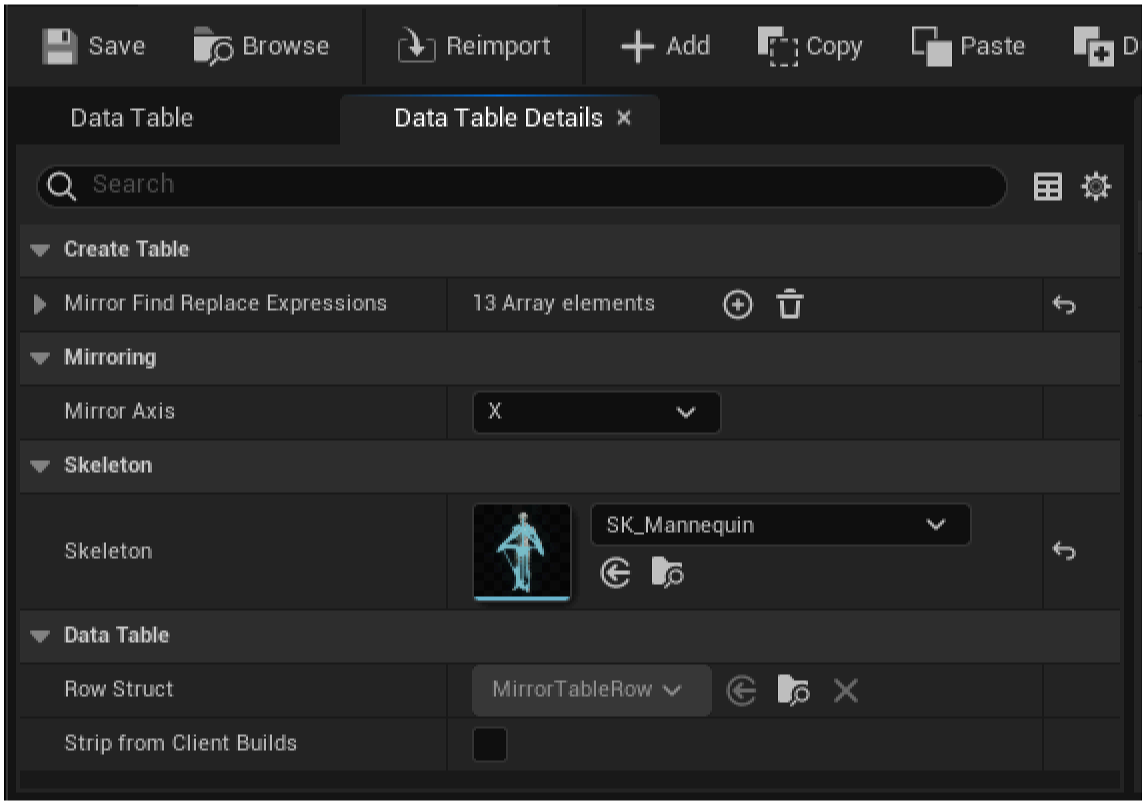
	Row Name	Name	Mirrored Name	Mirror Entry Type
1	root	root	root	Bone
2	pelvis	pelvis	pelvis	Bone
3	spine_01	spine_01	spine_01	Bone
4	spine_02	spine_02	spine_02	Bone
5	spine_03	spine_03	spine_03	Bone
6	spine_04	spine_04	spine_04	Bone

spine_01

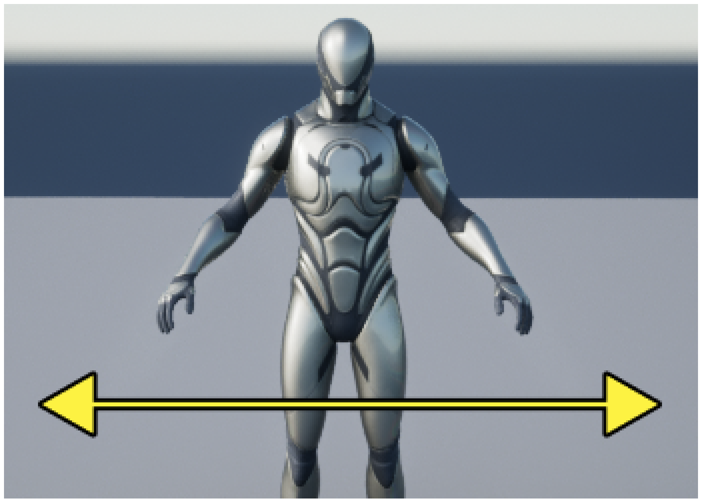
Name	spine_01
Mirrored Name	spine_01
Mirror Entry Ty...	Bone

Data Table Details

The Data Table Details panel contains configurations and other settings for the mirror behavior:



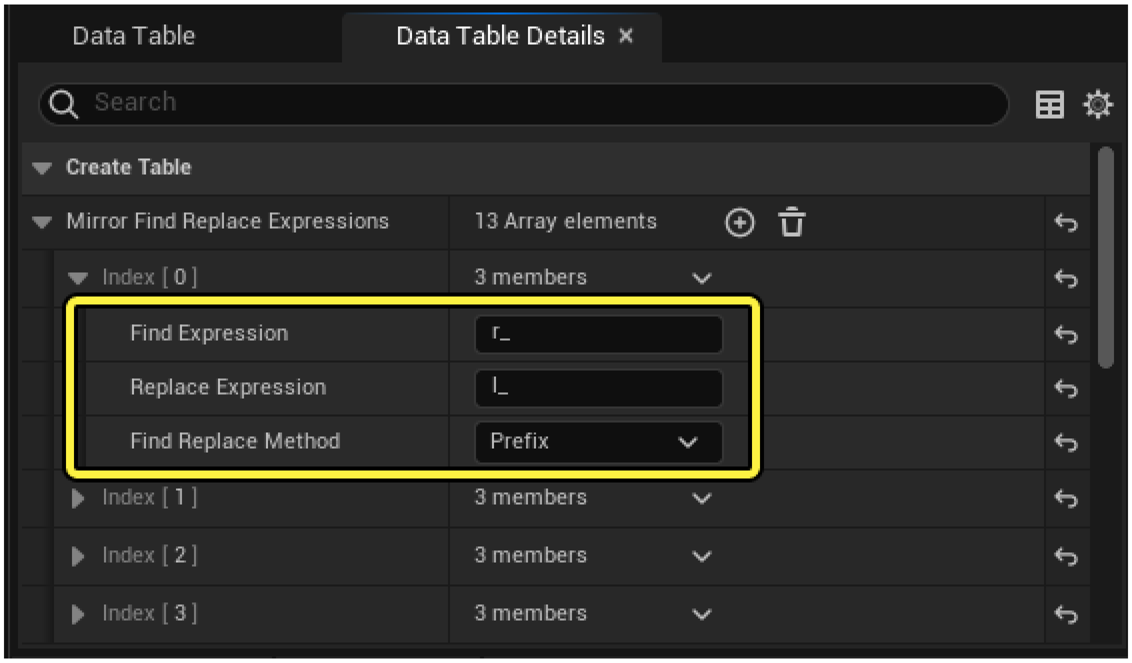
Create Table
Mirror Find Replace Expressions: 13 Array elements
Mirroring
Mirror Axis: X
Skeleton
Skeleton: SK_Mannequin
Data Table
Row Struct: MirrorTableRow
Strip from Client Builds

Name	Descriptions
Mirror Find Replace Expressions	An array of expressions that are used to automatically populate the table with common mirror entries. Refer to the Find and Replace Expressions section for more information.
Mirror Axis	The axis of mirroring, which is across the front of the character. In most cases this should be **X**. 
Skeleton	The Skeleton Asset to use for the mirror operation.

Name	Descriptions
Row Struct	The structure to use for each row of the table. You must inherit from `FTableRowBase` if you want to extend this.
Strip from Client Builds	Enabling this will not cook this Data Table into client builds. Useful if you are making confidential tables that only servers should know about.

Find and Replace Expressions

Find and **Replace Expressions** are array entries you can add to the Data Table Details that automatically search and replace certain string text of elements in the skeleton. These expressions are then used to inform which elements are automatically populated when creating or reimporting the skeleton.



Mirror Find Replace Expressions: 13 Array elements
Index [0]: 3 members
Find Expression: r_
Replace Expression: l_
Find Replace Method: Prefix
Index [1]: 3 members
Index [2]: 3 members
Index [3]: 3 members

Each array requires the following expressions:

Name	Description
Find Expression	The text to search for. In most cases this will likely be the symmetry modifiers for your element names, such as `_l`, `left_`, or `_left_`.

Name	Description
Replace Expression	The text to replace. In most cases this will likely be the symmetry modifiers for your element names, such as `_r`, `right_,` or `_right_`.
Find Replace Method	The search method to use when replacing text. You can select from the following options: • **Prefix**, which will search for text only at the beginning of the name. • **Suffix**, which will search for text only at the end of the name. • **Regular Expression**, where you can write a custom expression for finding and replacing names.

When using **Regular Expression** as your method, you can write custom Regular Expressions for **Find Expression** and **Replace Expression**.

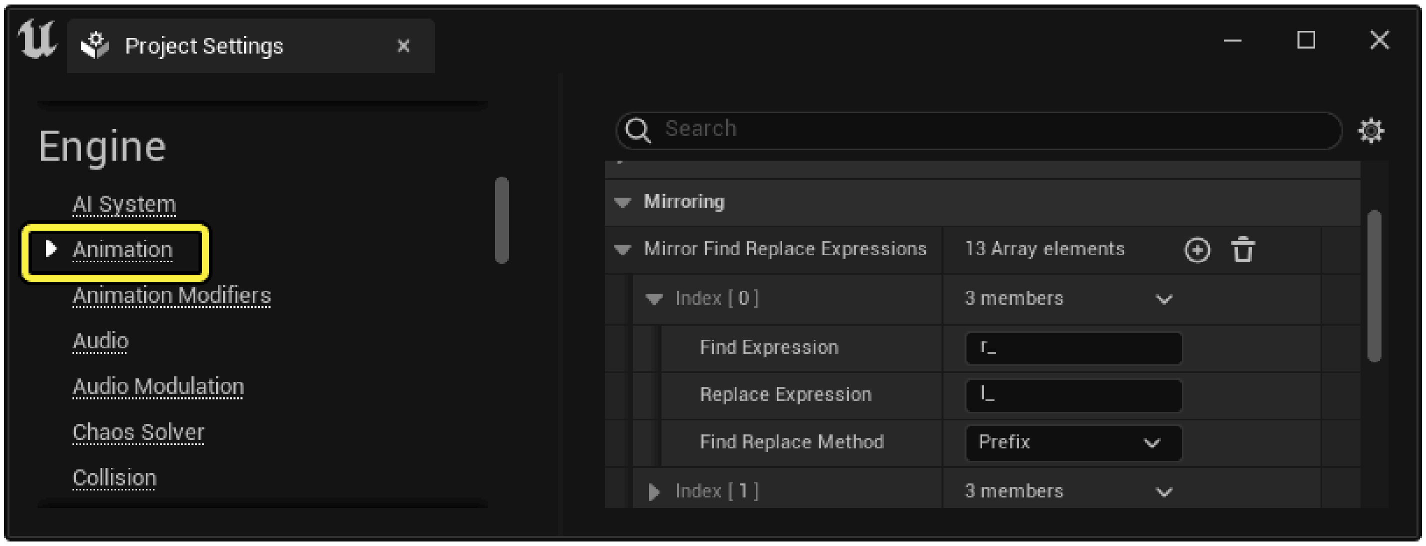
For example, **Index 12** of the default array contains the following expression, searching for common central bone names and assigning to the **Name** and **Mirrored Name** properties:

- Find Expression: `((?:^[sS]pine|^[rR]oot|^[pP]elvis|^[nN]eck|^[hH]ead|^ik_hand_gun).*)`
- Replace Expression: `$1`

In cases where your symmetry text modifiers are in the middle of an element name, such as `finger_left_index1`, you can write the following expression to correctly search and replace:

- Find Expression: `(\S*)_left_(\S*)`
- Replace Expression: `$1_right_$2`

The array comes pre-populated with common expressions, such as searching and replacing several permutations of symmetry modifiers as pre or postfixes. You can change this default array from the Project Settings. In the main Unreal Engine menu bar, select **Edit > Project Settings**, then navigate to the **Engine > Animation** section and locate the **Mirroring** property section.


Project Settings

Engine

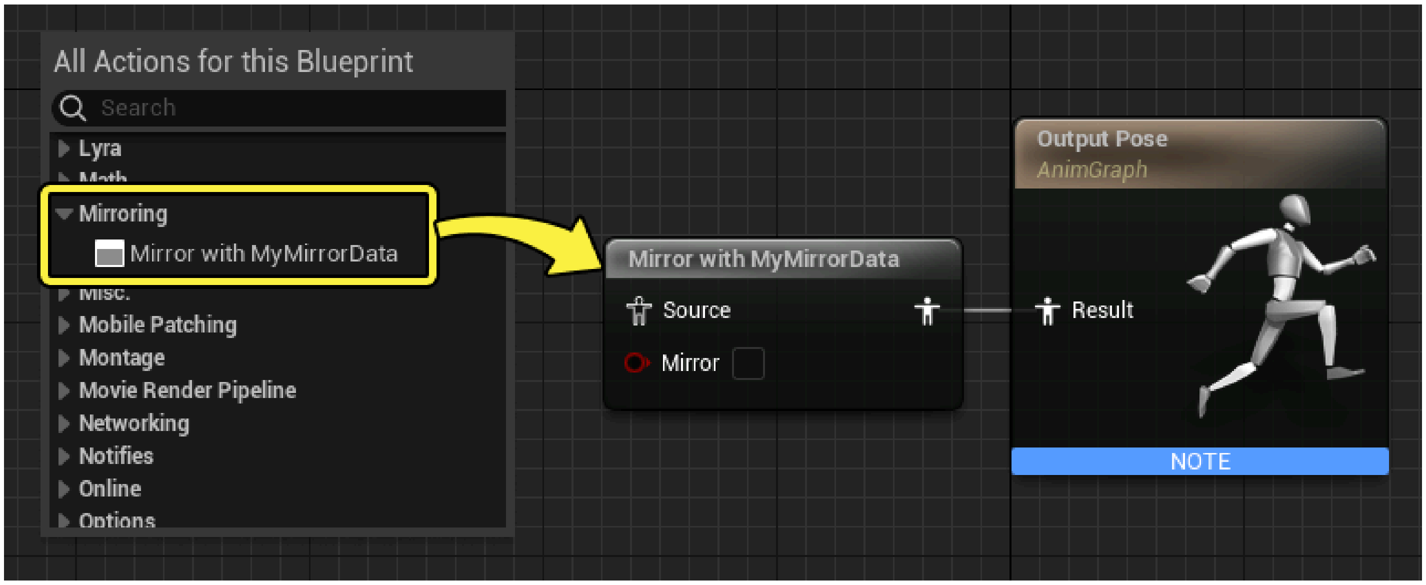
- AI System
- Animation
- Animation Modifiers
- Audio
- Audio Modulation
- Chaos Solver
- Collision

Mirroring
Mirror Find Replace Expressions: 13 Array elements
Index [0]: 3 members
Find Expression: r_
Replace Expression: l_
Find Replace Method: Prefix
Index [1]: 3 members

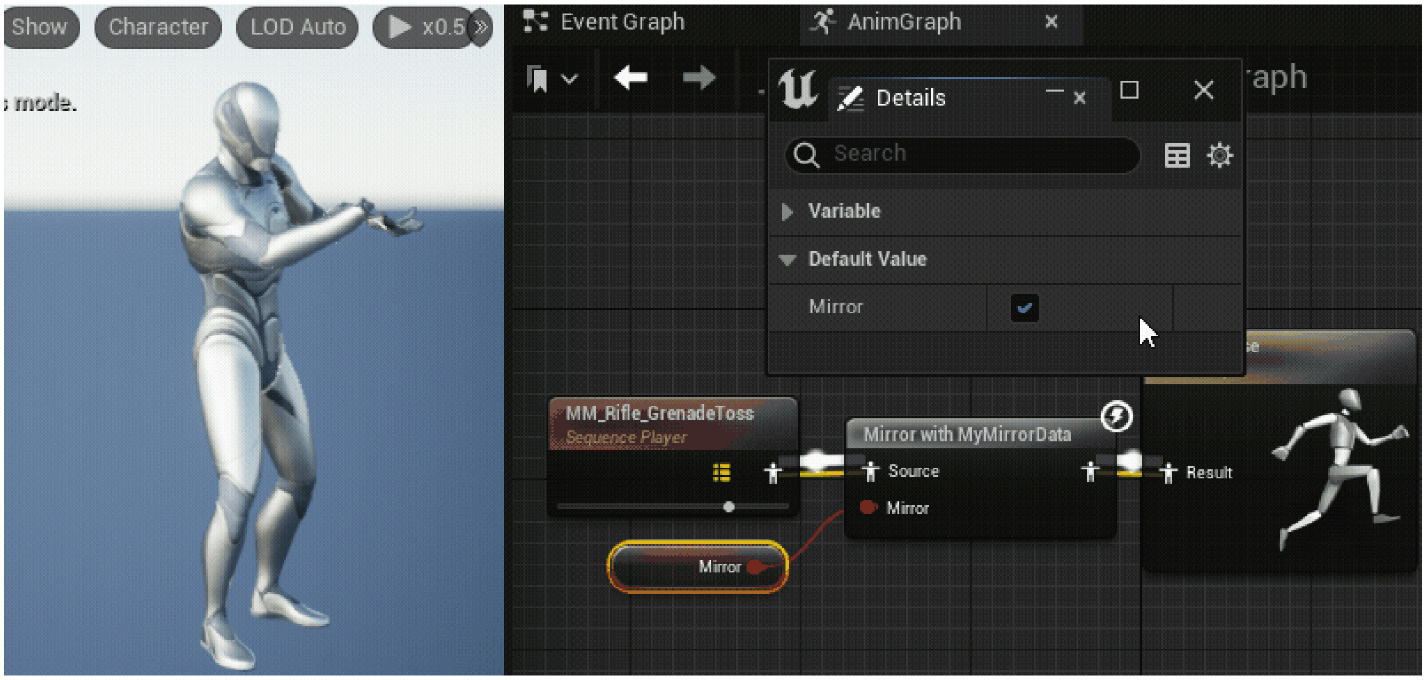
Mirroring Animation

Once you have created and populated your Mirror Data Table, you can now mirror your animations in Animation Blueprints. This is done using the **Mirror** AnimGraph node.

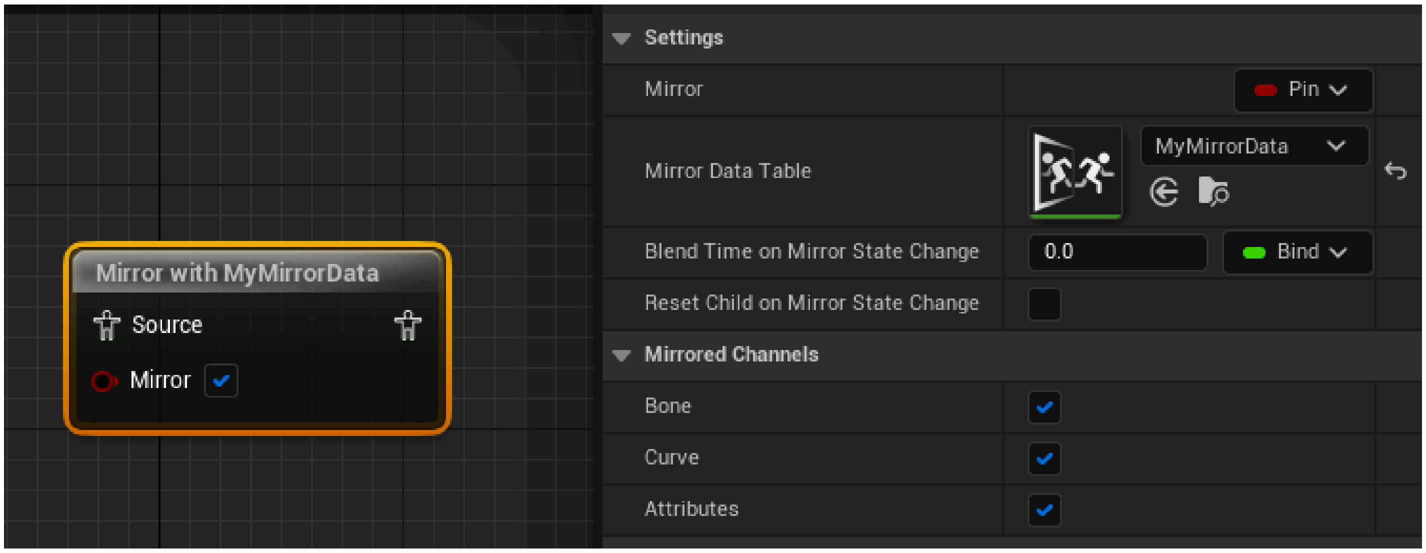
To create it, right-click in the AnimGraph and select your table from the **Mirroring** category.



You can preview the mirroring effect by providing an input pose and bool variable, enabling or disabling the mirroring.

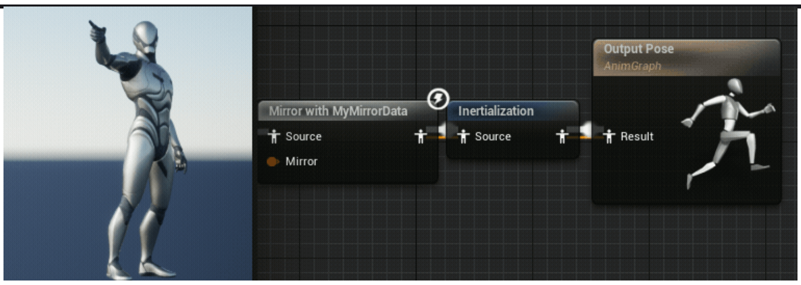


The Mirror node contains the following properties:



Settings	
Mirror	Pin
Mirror Data Table	MyMirrorData
Blend Time on Mirror State Change	0.0 Bind
Reset Child on Mirror State Change	
Mirrored Channels	
Bone	✓
Curve	✓
Attributes	✓

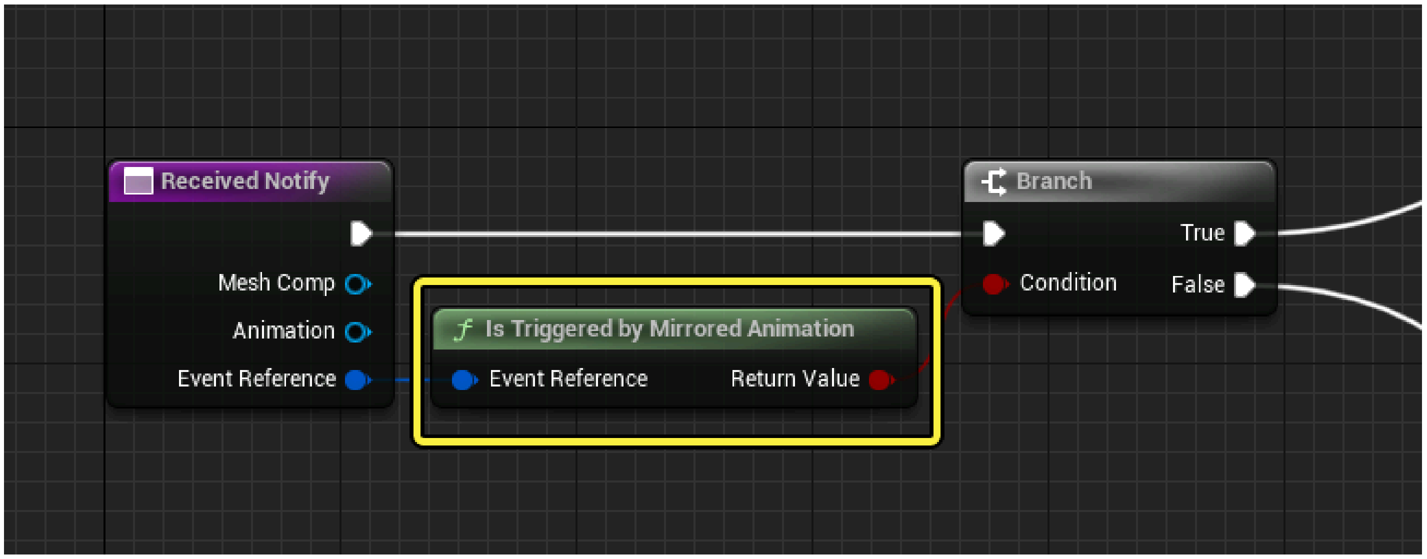
Name	Description
Mirror	Enables or disables the mirror effect. This is exposed as a pin by default.
Mirror Data Table	The Mirror Data Table to use for mirroring.
Blend Time on Mirror State Change	The length of time to blend between mirror states when **Mirror** is enabled or disabled. Using this also requires you to use an Intertialization node after the Mirror node.

Name	Description
	 Mirror with MyMirrorData Source Mirror Inertialization Source Output Pose AnimGraph Result
Reset Child on Mirror State Change	Enabling this will reinitialize the source pose when the mirror state changes.
Bone	Whether or not to include bone data in the mirroring.
Curve	Whether or not to include curve data in the mirroring.
Attributes	Whether or not to include notify and sync marker data in the mirroring.

Detecting Mirrored Animation in Blueprint Notifies

If you are using Custom Notify States, then you may want to change it to have a different behavior based on the mirrored state. You can differentiate between mirrored states in the Notify Blueprint using the **Is Triggered By Mirrored Animation** node.

In this example, it is being used in the **Received Notify Function** to branch the logic, checking if the notify was received from a mirrored animation or not.


Received Notify
Mesh Comp
Animation
Event Reference
Is Triggered by Mirrored Animation
Event Reference
Return Value
Branch
True
Condition
False