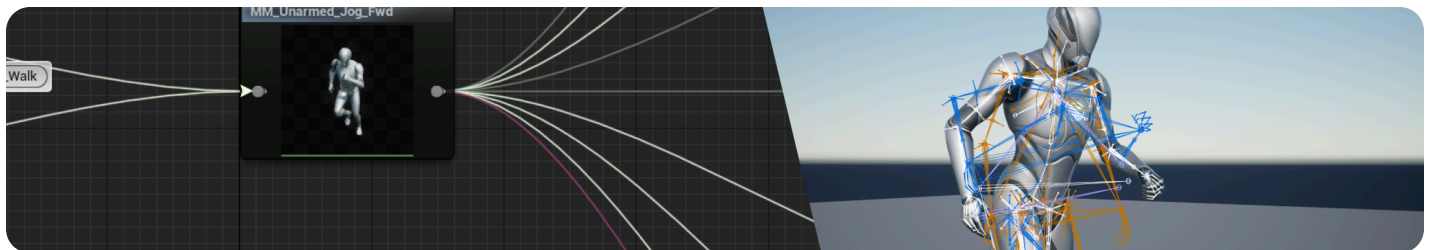


- Developer
- / Documentation
- / Unreal Engine ▾
- / Unreal Engine 5.4 Documentation
- / Animating Characters and Objects
- / Skeletal Mesh Animation System
- / Animation Shortcuts and Tips

Animation Shortcuts and Tips

Workflow shortcuts and tips for animators and animation programmers in Unreal Engine.



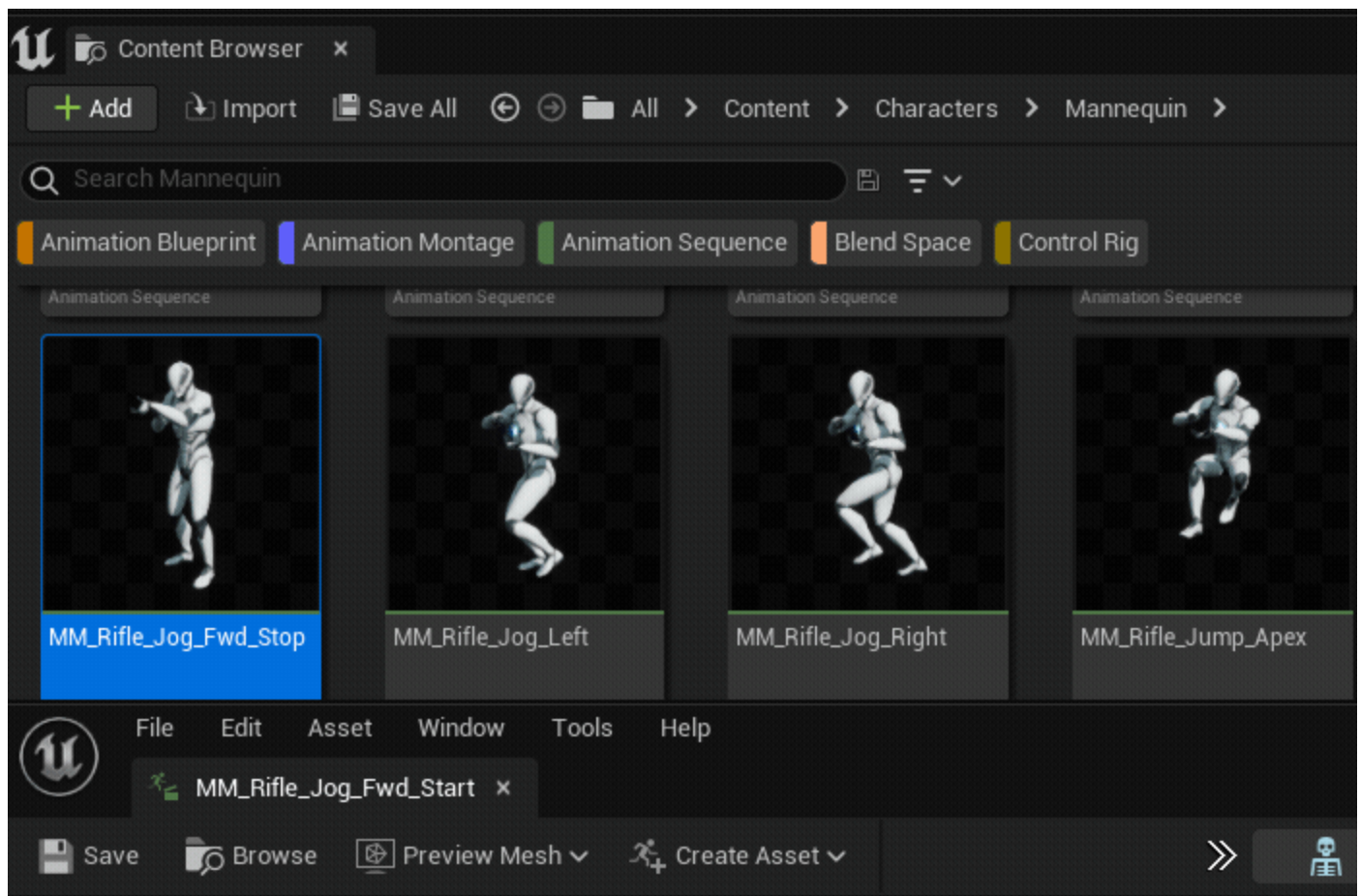
Unreal Engine animation workflow shortcuts and tips for animators and animation programmers.

Navigating Assets

Here you will find workflow shortcuts and tips to assist in navigating and working with [Animations Assets](#).

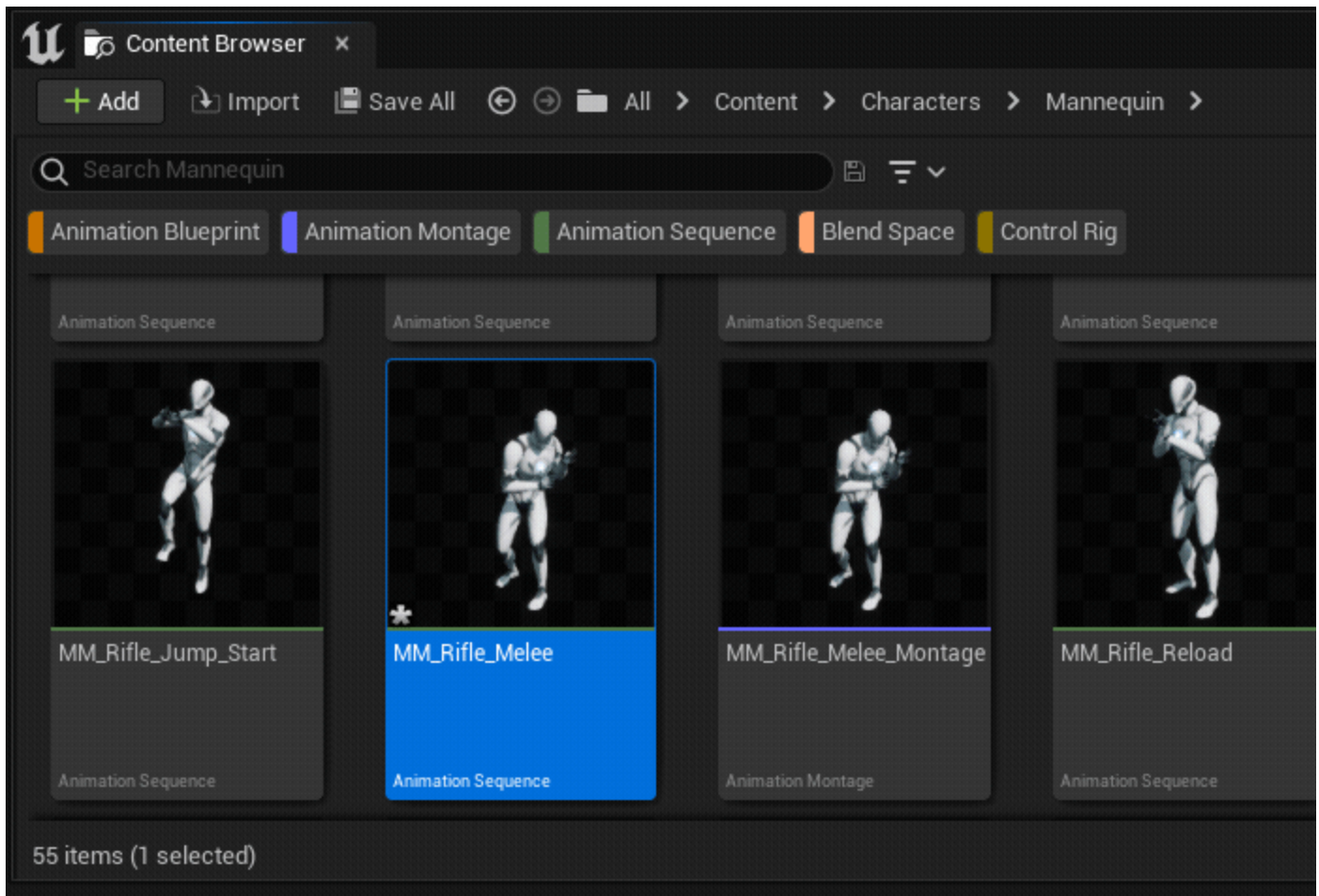
Opening in Separate Tabs

In Unreal Engine, opening a second animation asset of the same type overrides the open asset. To open multiple animations assets of the same type, hold **Shift** while clicking on another asset to open it in a new adjacent tab.



Revert Unsaved Changes

Unsaved assets will be indicated with an asterisk (*), in the [Content Browser](#). You can revert these unsaved changes by right-clicking the asset and selecting **Asset Actions > Reload** from the context menu.



Content Browser Filtering

The Content Browser has various [filters](#) for all file types, including animation content. The **Other Filters** category also contains helpful filter options, such as **Checked Out** files when working within a server-based development environment.

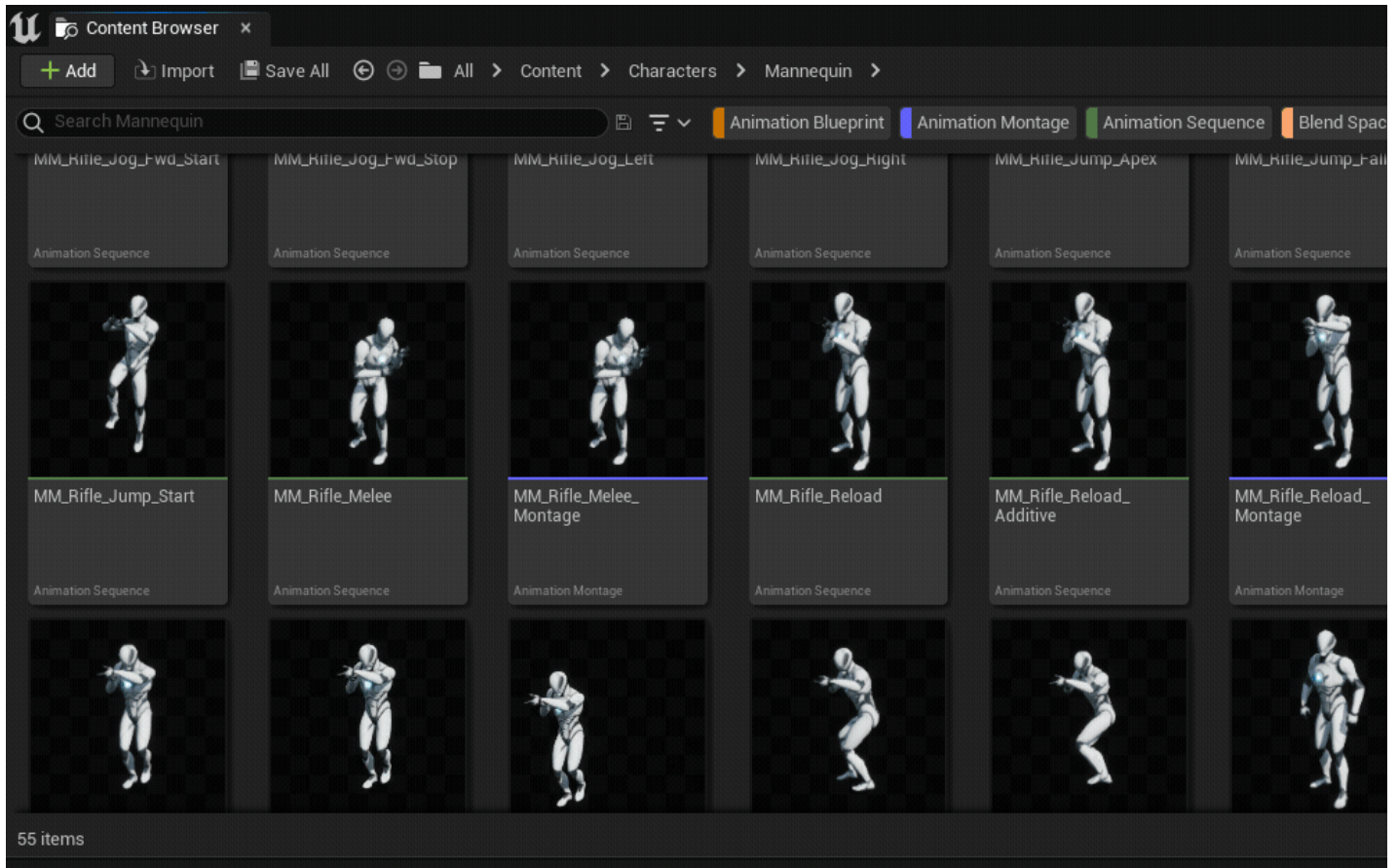
Animation Asset Filtering

With an Animation Asset open, you can filter for content in the [Asset Browser](#) tab. Especially useful filters include **Anim Filters > Uses Curve** for finding animations that rely on an [Animation Curve](#) and **Anim Filters > Uses Skeleton Notify** for animations containing [Animation Notifies](#).

Reference Viewer

The **Reference Viewer** is a tool you can use to visually see where assets are being used in your project. This is especially useful with animation assets to reference which [Animations Sequences](#) or [Montages](#) are being referenced in an [Animation Blueprint](#).

First select the asset you wish to see in the Reference Viewer. Then, use the hotkey combination **Alt + Shift + R** to open the Reference Viewer window.



Content Browser Advanced Search Syntax

Refer to the [Advanced Search Syntax](#) documentation for information on how to use the Content Browser's advanced search features.

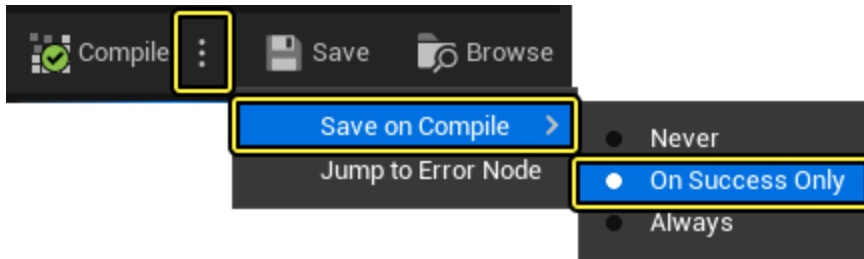
Animation Blueprints

In the following section you will find helpful Tips and Tricks to aid in the navigation and editing of [Animation Blueprints](#).

Compile Settings

When working with Animation Blueprints, compiling and saving are frequent actions that are preformed in tandem. To enable these functions to occur simultaneously, first open the breakout menu next to **Compile** in the **Toolbar**, indicated by the three horizontal dots.

Within this menu, navigate to the **Save On Compile** option and choose **On Success Only**.



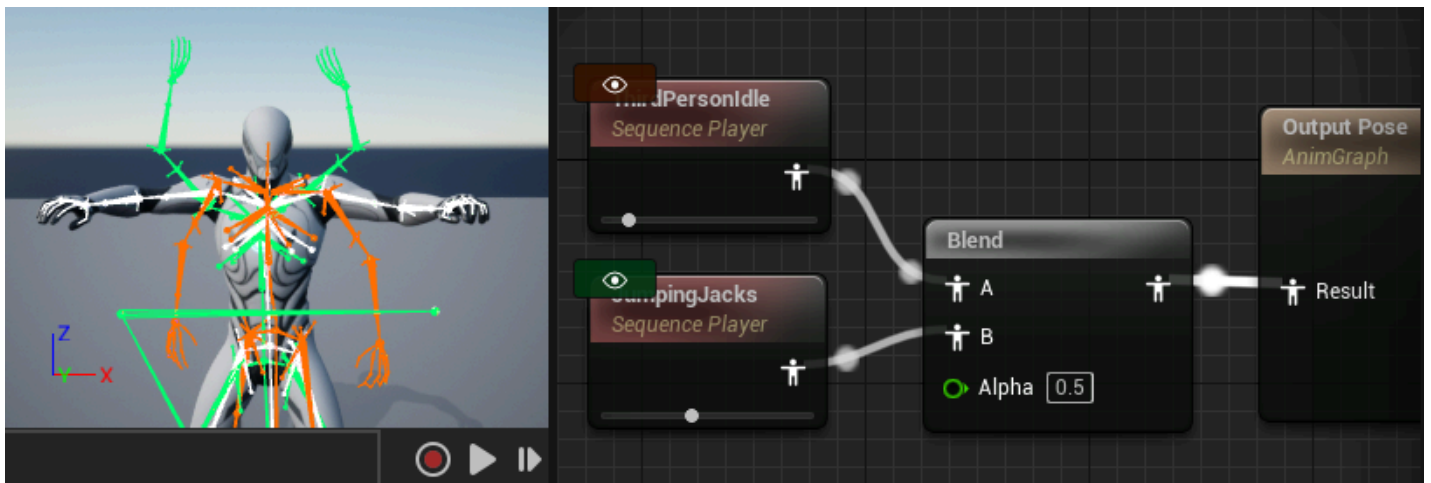
For debugging purposes, the **Jump To Error Node** option in the **Compile** setting breakout menu, will locate and show the point of error or warning in the Animation Blueprint, when a compile has been unsuccessful or produces a warning.

Blueprint Editor Cheat Sheet

Refer to the [Blueprint Cheat Sheet](#) for useful general tips on working with the Blueprint Editor.

Pose Watch

Right-click any [Animation Blueprint Node](#) in the **Animation Graph** and select **Toggle Pose Watch** to see a debug drawing of the pose during animation preview playback.



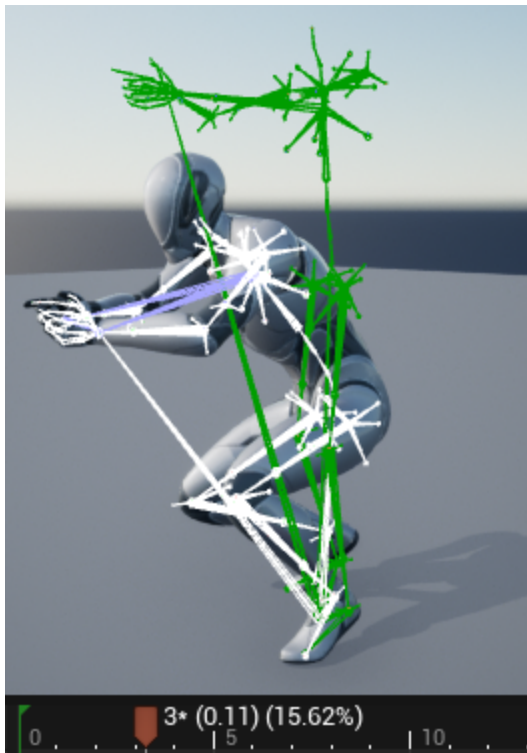
Toggle multiple animation nodes to see all animations in effect. You can manage the pose watch toggled animations and adjust their properties, such as color, in the **Pose Watch Manager** window. To open this window, navigate to the **Window** option in the **Menu Bar**, and select **Pose Watch Manager** in the **Animation Blueprint Editor** section.

Animation Sequence and Montage Editor

In this section you will learn helpful tips and tricks about working within the [Animation Sequence Editor](#).

Visualizing Additive Animations

When previewing [Additive Animations](#) in the viewport, click the **Character** tab and open the **Animation** breakout menu. Under the **Animation** header select the **Additive Base** option to overlay the base pose skeleton over the Additive Animation.



Editor Preferences

Below is a list of helpful properties you can adjust in the **Editor Preferences** window. You can open the Editor Preferences window in the **Menu Bar** by selecting **Edit > Editor Preferences**.

AutoSave

You can enable or disable the Unreal Engine's **AutoSave** feature with the **Enable AutoSave** setting. In the **Editor Preferences** window, navigate in the sidebar to **General > Loading & Saving**, you will find the AutoSave section, which contains the **Enable AutoSave** property.

Level Loading

In the **Editor Preferences**, set the **Load Level at Startup** property to **Last Opened**, to reopen the last level open in the editor when relaunching Unreal Engine.

Play In Editor (PIE)

In the section below you will find a reference guide of helpful commands to enter into the **Console** while running or simulating a project in the Editor. While simulating a project, press the **Backtick** (```) key to bring up the **Console** command line. Type a command then press **Enter** to execute it within the Console.



While debugging, you may find yourself frequently running the same commands. You can send recently used commands to the Console by pressing the **Up Arrow** key after initializing the command line to bring up a list of recently used commands, navigate to the desired command using the **Arrow** keys and press **Enter** to reactivate it.

Animation Debug Text

Command	Description
NextDebugTarget (PGUP)	Cycles to the next debug target.

Command	Description
PreviousDebugTarget (PGDOWN)	Cycles to the previous debug target.
ShowDebug	Toggles the display state of all applicable debug features.
ShowDebug ANIMATION	Toggles display the state of animation debug data.
ShowDebugToggleSubCategory	Toggles display of a specific category. Categories will appear as autocomplete results.

Miscellaneous Commands

Command	Description
a.animnode."NODE_NAME"	Provides debug options for various animation nodes.
Log <input type="text" value=" <Category> <Verbosity>"/>	Changes log verbosity. Example: <input type="text" value=" Log LogAnimMontage Verbose"/>
p.VisualizeMovement 0	Hides the movement component debug.
p.VisualizeMovement 1	Shows the movement component debug.
show bones	Toggles the rendering of the project's objects' bones.
show collision	Toggles the rendering of the project's objects' collision.
Slomo 0.5	Slows the speed at which the game is displayed. The numerical value is the factor to which the game is slowed. Example: 0.5 = Half Speed.

Command	Description
Stat FPS	Toggles the display of the project framerate in frames per second.
t.MaxFPS 0	Sets the project's framerate limit. A value of 0 will remove the framerate limit.



Framerate adjustments affect the Editor and can have detrimental effects in the functionality of the editor on weaker hardware.

Debugging LODs

Command	Description
a.VisualizeLODs 0	Hides displayed LOD info.
a.VisualizeLODs 1	Displays LOD info.
FORCESKELLOD LOD=0	Forces all Skeletal Meshes to render at specified LOD level. For example: FORCESKELLOD LOD=0 will render LOD 0, LOD=1 will render LOD 1, ext.
FORCESKELLOD LOD=-1	Disables any forced LOD rendering.

Debugging Properties

Command	Description
DisplayAll <ClassName> <PropertyName>	Displays the value of a property on all objects of a specific class.

Command	Description
	<p>Example: DisplayAll CharacterMovementComponent Velocity, displays the value of the velocity of all character movement components in the level.</p> <p>Example: DisplayAll MyAnimBP_C AimYaw, displays the value of AimYaw for all Animation Blueprints in the class.</p>
DisplayClear	Clears the displayed results of a DisplayAll command.
GetAll <code><ClassName> <PropertyName></code>	Possesses the same functionality as the DisplayAll command, but prints the values in the Output Log rather than displaying debug text in the editor.
Display <code><ObjectName></code> <code><PropertyName></code>	Displays the value of a property for a single defined instance.

When using the **Display** commands to display property values, the **GetAll** command can be useful for identifying which actors to define in the comand's `<ObjectName>` field.

Identify the desired Actor's instance in the [World Outliner](#). For example: `BP_MyPawn_C_3`

Run: `GetAll MyAnimBP AimYaw`



Find the path to the object's name in the **Output Log**. For example:

`/Temp/UEDPIE_0_Untitled_1.Untitled_1:PersistentLevel.BP_MyPawn_C_3.CharacterMesh0.MyAnimBP_C_0`

Run: `Display`

`/Temp/UEDPIE_0_Untitled_1.Untitled_1:PersistentLevel.BP_MyPawn_C_3.CharacterMesh0.MyAnimBP_C_0 AimYaw`

Memory Tracking

Command	Description
obj list class="AnimSequence"	Lists all animation sequence assets that are loaded. Executing this command is a recommended procedure in a cooked build to ensure all assets are present.
obj refs name="ASSET_NAME"	<p>Prints the reference chain of the specified asset.</p> <p>Example: <code>obj refs name=</code> <code>/Game/Characters/Animations/ThirdPersonJump_End.ThirdPer</code> <code>sonJump_End</code></p>

Cheat Scripts

Use **Cheat Scripts** to combine multiple **Console** commands into a single command. You can define and set up Cheat Scripts by adding them to your project's **DefaultGame.ini** file, found in the projects **Config** folder.

Example:

```

1 [CheatScript.ShowAnimVars]
2 +Cmd="DisplayClear"
3 +Cmd="DisplayAll CharacterMovementComponent Velocity"
4 +Cmd="DisplayAll MyAnimBP_C AimYaw"
5

```

 Copy full snippet

The new executable command `ShowAnimVars`, will now perform the function of `DisplayClear`, `DisplayAll CharacterMovementComponent Velocity`, and `DisplayAll MyAnimBP_C AimYaw` in succession.

Editor Utility Widgets

[Editor Utility Widgets](#) allow custom editor widgets to be created entirely in Blueprints. A common use case is to create a set of buttons that trigger common **Console** commands.