Skip to content

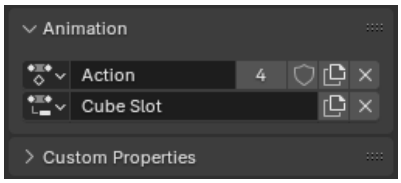
Actions

Actions are Blender's container for animation data. For example, when you animate the location of an object, that animation is stored in an action rather than directly on the object itself. The object then uses the action to get animated, much the same way that a mesh uses a material to get shaded. Al animatable data-blocks (objects, meshes, materials, etc.) are animated this way: they don't store their own animation data, but instead use an action that stores the animation data for them.

Actions are also data-blocks themselves, and therefore can be easily appended or linked into other blend files. This lets actions be used not just for storage, but also for organizing and reusing animation data. For example, if you're building a library of animations (run cycles, jumps, idling, etc.), each animation can go into its own action, which can then be conveniently linked or exported as a distinct animation.

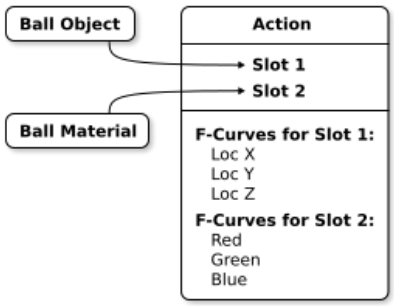
Action Slots

The animation data inside an action is further organized into *Slots*. Each action has a set of slots and different animation data for each of those slots. An animated data-block then specifies both an action and a slot within that action, and that determines which animation data the data-block is animated by.



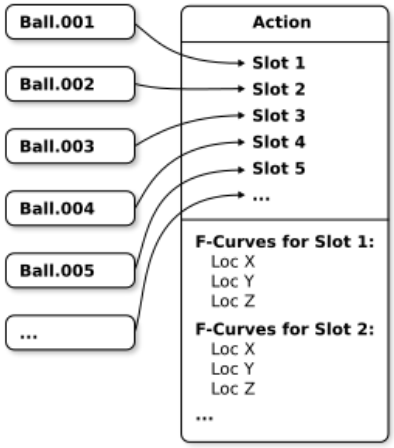
Action selector and its accompanying slot selector in the properties of an object, for seeing and selecting which action and slot animate the object.

The purpose of slots is to allow an action to store distinct animation data for multiple data-blocks. For example, you may have an animation of a bouncin ball that changes its color on each bounce, and that involves two data-blocks: the object and its material. Slots allow you to put both the object's animati and the material's animation in the same action by having a different slot for each.



Visualization of a ball and its material connected to different slots in an action.

In this example there is one slot for an object and one slot for a material, but you can have as many slots as you like for as many objects, materials, lights etc. as you like. If you're baking down a simulation of 100 bouncing balls, you could store that animation in single action with 100 slots.



Visualization of many balls all connected to different slots in an action.

Not all actions need to take advantage of slots: you are free to use 100 separate actions for all those bouncing balls if you prefer. Nevertheless, the animation data in an action is always organized into slots, and therefore you need at least one slot in an action in order to animate something.

Note that slots are not "for" any specific data-block: any data-block can use any slot. For example, you can have two different characters use the same slot in the same action, and they'll both simply get animated by the same animation data. Slots are just a way to organize distinct animation data within an action, and don't have any intrinsic attachment to anything in the scene.

Note

Internally, the animation data in an action is further organized into layers and strips. This is not currently exposed in the UI and does not impact how you use actions right now. It is purely in preparation for future animation features that are not yet in Blender, and you can safely ignore it for now.

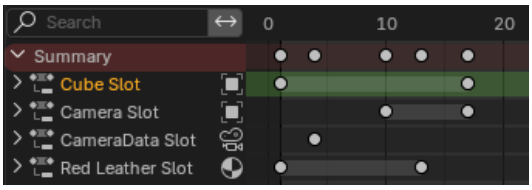
However, layers and strips *are* exposed in the Python API, so you will need to be aware of this when writing scripts and addons that work with actions. See the Python API documentation for more details.

Slot Names and Associated Types

Each slot in an action has a name, and you are free to name them whatever you like. By default, new slots are named after the last slot assigned to the data-block they were created for, or after the data-block itself if it's never been assigned a slot before.

In addition to having a name, each slot also has an associated data-block type that it is intended for (for example, "material", "object", etc.). This is set automatically when a slot is first assigned to animate a data-block.

One of the places you can see a slot's associated type is in the action editor's channel list, where it's displayed as an icon next to the slot's name.



Slots displayed in the Action Editor's channel list, with their associated type as an icon to the right of their name.

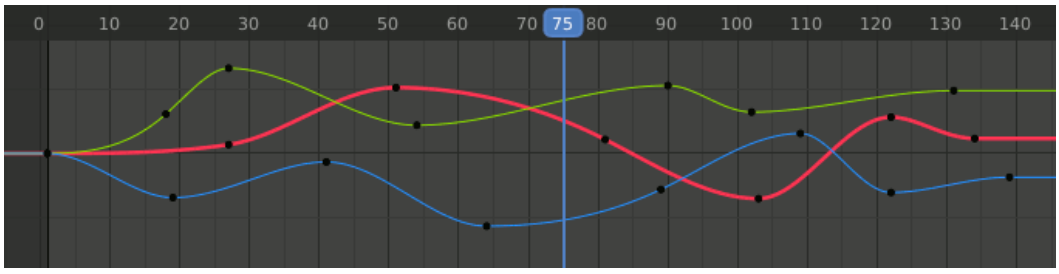
Within an action, a slot must have a unique combination of name + associated type. For example, you can have two slots named "Cube" in an action as long as one of them is for objects and the other is for materials, but not if they are both for objects. When they are both for objects, their associated type the same, and thus they must have different names. In that case Blender will use the familiar approach and name them "Cube" and "Cube.001".

Note

Although it's not useful, and Blender makes this difficult to do, it is nevertheless possible to cause slots to get assigned to a data-block of the wrong type. For example, assigning a slot intended for materials to an object. Nothing bad happens if you manage to do this, but the F-Curves of that slot are unlikely to match any properties on the mismatched data-block, and therefore won't animate anything.

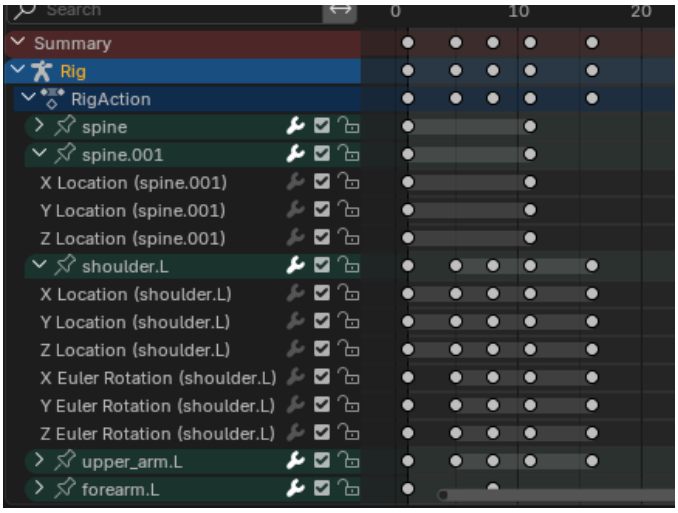
F-Curves & Channels

F-Curves are the fundamental unit of animation in Blender, and are the main kind of animation data that actions contain. Each F-Curve contains keyframes that define how a property (such as the X location of an object) should change over time.



Graph Editor, displaying three F-Curves for three different properties.

Blender's animation editors (such as the dopesheet, graph editor, etc.) have a **channel list** on their left side that display animated properties. For actions these channels correspond to the F-Curves that animate those properties.



The Dopesheet Editor's channel list, with the animated channels of various bones grouped under their bone names.

Channels also support a limited form of organization called "channel groups". For example, by default Blender creates a channel group for the channels c each bone. There are a few features in Blender that rely on the groups, but mostly they are just for your convenience.

Working With Actions

When you first animate an object (or other data-block) in Blender, Blender tries to automatically find an appropriate action for it, or if it can't find an appropriate action then it will create one. After an action has been assigned, it also creates and assigns a new slot for the data-block.

Blender uses heuristics to try to find an appropriate action, based on the idea that animation of closely related data-blocks should typically go in the same action. For example, an object and its data are considered closely related, so if a camera object is already animated and you insert keys for its focal leng (which lives on the camera data, *not* the camera object), the action that's assigned to the object will be reused for the camera data as well. These relationships go both ways, so the action will also be reused when keying the camera object if the camera data is already animated.

Some examples of other data-blocks that are considered closely related for this purpose are: materials and their embedded node trees, worlds and their embedded node trees, and meshes and their shape key data.

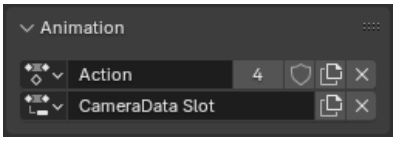
Note

There is an exception to this "closely related" heuristic, which is when a data-block has more than one user. For example, if a single mesh data-block is used by multiple mesh objects, then the relationship is ignored and the mesh data and its users will get separate actions despite otherwise being considered closely related.

Manually Assigning Actions and Slots

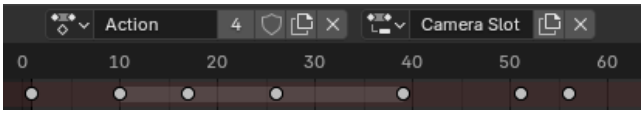
In addition to letting Blender automatically choose an action and slot for a data-block, you can also manually assign them. This can be used to assign existing animation to a data-block by selecting both the action and slot. It can also be used to specify an action for a data-block's keys to go into, by assigning the action but leaving the slot blank, in which case a new slot will be created when the first key is set.

For each data-block in the properties editor there is an Animation panel with action and slot selectors. You can use these to assign actions and slots to a data-block.



The action and slot selector for Camera data in the Properties Editor.

For the active object you can also assign its action and slot in the action Editor's header.



The action and slot selector for the active object (in this case a camera object) in the Action Editor.

The action and slot selector for the active object (in this case a camera object) in the Action Editor.

See also

- Scene Animation Panel
- Speaker Animation Panel
- Movie Clip Animation Panel
- Mask Animation Panel

Note

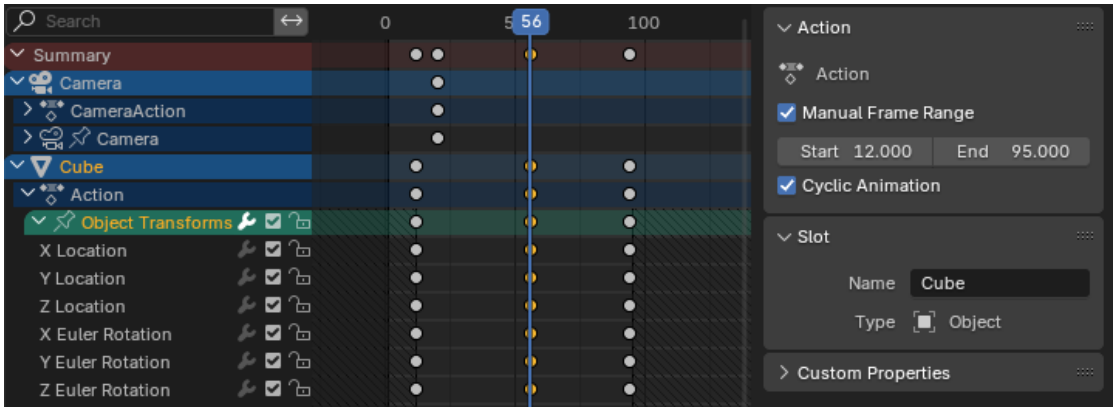
When selecting a slot for a data-block, you won't necessarily see all the slots of an action listed in the dropdown. This is because Blender limits that dropdown to the slots with an associated type that matches the data-block.

When you select an action to animate a data-block, for convenience Blender attempts to automatically select an appropriate slot for you based on name and associated type. If no appropriate slot is found then the slot selector will remain empty, in which case you can manually select an existing slot, create new one, or just start keying and let Blender automatically create a new slot for you. If Blender assigns a slot you didn't want, you can select another slo manually or simply clear the slot selection.

NLA

Actions can also be assigned to NLA strips within a data-block's NLA system. Please see the documentation for the NLA Editor for how to animate data-blocks via the NLA system.

Action Properties



Actions with and without a Manual Frame Range in Dope Sheet.

It is possible to manually specify the intended useful frame range of an action via a panel available in the Dope Sheet or the NLA Editor when a channel o NLA track is selected.

Manual Frame Range

Manually specify the intended playback frame range for the action (this range is used by some tools, but does not affect animation evaluation). The manual frame range feature can be toggled with the checkbox.

When the range is set, it is used instead of the actual range occupied by key frames when adding a new track based on the action to NLA. It can also be used by exporters to determine the range of frames to export.

The range is displayed in the background of the editor as diagonal hash fill, to distinguish it from the solid fill of the current playback range.

The frame values are most commonly expected to be integers, but can be fractional.

Cyclic Animation

Specifies that the action is intended to be cyclic over the specified range. The first and last frames of the range should represent the same pose of t cycle one loop apart, i.e. the range should include the duplicated initial key of the loop.

Note

This option signifies intent and does **not** make the action cycle on its own. However, if Cycle-Aware Keying is enabled, it will automatically enable cyclic extrapolation and set up the loop period for curves newly added to the action.

Slot

The properties of the action slot that is used by the currently selected item in the channel list.

Name

The name of the slot.

Type

The data-block type that the slot is intended to animate.

Custom Properties

Create and manage your own properties to store data in the action's data block. See the Custom Properties page for more information.

Previous
Shrinkwrap Constraint

N
Driv

Made with Furo
Last updated on 2025-05-10

View Source
View Translation
Report issue on this page