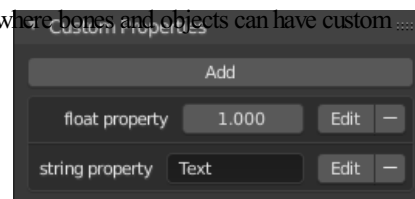# Custom Properties

Custom properties are a way to store your own data in Blender's data-blocks. It can be used for rigging (where bones and objects can have custom properties driving other properties), and Python scripts, where it's common to define new settings not available in Blender. It is also possible to access custom properties from materials via the Attribute Node.

Only certain data supports custom properties:

- All data-blocks types.
- Bones and pose bones.
- Sequence strips.

Custom Properties panel.

To add a custom property, search for the *Custom Properties* panel, found at the bottom of most Properties or Sidebar region, and click *New*. Properties can be removed from the same location with the delete icon. Once properties are added they can be configured via the edit icon to work for a particular use case; see Editing Properties for more information.
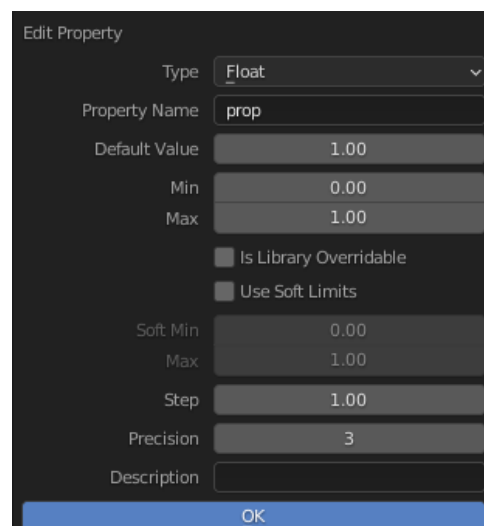
Editing Properties

User Interface

Custom properties can be edited using the panel available for data types that support it. Editing the properties allows you to configure things such as default values, ranges, and even add a custom tooltip.

Custom Properties edit pop-up.

Type

The data type of the property; different data types have can only have specific data properties.

Float:

A numeric value with decimals e.g. 3.141, 5.0, or 6.125.

Float Array:

A collection of multiple float data types e.g. `[3.141, 5.0, 6.125]`. This data type can also be used for data that can be represented as a float array such as colors. These special float arrays can be set in the *Subtype* selector.

Integer:

A numeric value without any decimals e.g. 1, 2, 3, or 4.

Integer Array:

A collection of multiple integer data types e.g. `[1, 2, 3, 4]`.

Boolean:

A data type that has two possible values e.g. `True` or `False`.

Boolean Array:

A collection of boolean values e.g. `[True, False, True]`

String:

A sequence of characters such as "Some Text".

Data-Block:

A reference to a Blender object, see Data-Blocks.

Python:

Edit a Python data type directly, used for unsupported data types.

Array Length

The number of elements in the array. Note that if the array length is greater than 7 you cannot directly edit its elements, you must press *Edit Value* to edit the elements of the array.

Property Name

The text that is displayed to the left of the value. This name is also used to access the property via Python.

Default Value

This sets the default value of the property used by the Reset to Default Value operator.

> Warning
>
> Default values are used as the basis of NLA blending, and a nonsensical default (e.g. 0 for a property used for scaling) on a property intended for being keyframed is likely to cause issues.

Min, Max

The minimum/maximum value the custom property can take.

Library Overridable

Allow the property to be overridden when the data-block is linked.

Soft Limits

Enables limits that the *Property Value* slider can be adjusted to without having to input the value numerically.

Soft Min, Max

The minimum/maximum value for the soft limit.

Step

A multiplier to control how much the data type is incremented at a time. The internal step size for floats is 0.01, so a *Step* value of 5 will increment a rate of 0.05 and a *Step* value of 100 will increment by 1.0. For integers the internal step size is 1.

Precision

The number of digits after the decimal to display in the user interface for float data types.

Subtype

Specifies the type of data the property contains, which affects how it appears in the user interface. This option is only available for float properties and has different options for regular floats and float arrays. Note, the unit often depends on the Scene Units.

For regular floats:

Plain Data:

Data values do not have any special behavior.

Pixel:

A measure digital image resolution.

Percentage:

The displayed value is a percentage, typically you will want the Min and Max values to be 0 and 100.

Factor:

A percentage between an upper and lower bound which typical have a numerical significance.

Angle:

A measure between intersecting lines.

Time:

Time specified in seconds.

Distance:

Measure of space between items.

Power:

Work as a factor of time, measured in watts. This is used in Blender to measure light intensity.

Temperature:
> Intensity of heat present.

Wavelength:
> The distance between cycles of a wave measured in millimeters (mm), micrometers (µm), nanometers (nm), or picometers (pm).

For float arrays:

Plain Data:
> Data values do not have any special behavior.

Linear Color:
> Color in linear color space.

Gamma-Corrected Color:
> Color in gamma corrected color space.

Euler Angles:
> Euler Rotation angles.

Quaternion Angles:
> Quaternion Rotation angles.

> **Note**
>
> For either of the color subtypes to work as expected the *Property Value* must be a vector with three or four values depending on the availability of an Alpha Channel.

ID Type Data-Block
> The ID-block type. For example: Key, Image, Object, Material. See Data-Block Types for a full list.

Description
> Allows you to write a custom Tooltip for your property.

Python Access

Custom properties can be accessed in a similar way to dictionaries, with the constraints that keys can only be strings, and values can only be strings, numbers, arrays of such, or nested properties.

See the API documentation for details.