

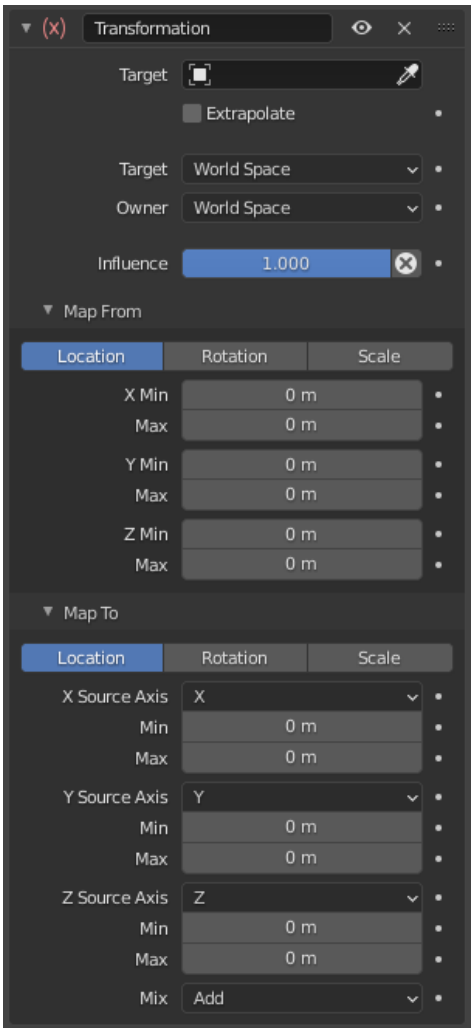
[Skip to content](#)

Transformation Constraint

This constraint is more complex and versatile than the other “transform” constraints. It lets you set the location, rotation or scale of an object/bone based on the location, rotation or scale of another, mixing and matching axes as you see fit. An example could be to set a gear’s X rotation based on the coordinate of a rail next to it.

The constraint works with input and output value ranges, one for each axis. It first clamps the input value to the *Map From* range, then offsets and scales to the corresponding *Map To* range. This lets you, say, map a Y coordinate in the range (-3m, 3m) to an X rotation in the range (0, 180°).

Options



Transformation panel.

Target

The reference object to read a transformation property from. If you don’t select one, the constraint’s icon will turn red and it will have no effect.

See [common constraint properties](#) for more information.

Bone

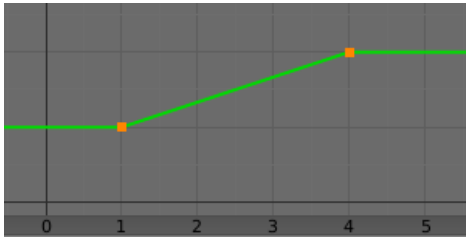
If *Target* is an [Armature](#), you can optionally choose a bone here to use the transformation of that bone instead of the armature object as a whole.

Extrapolate

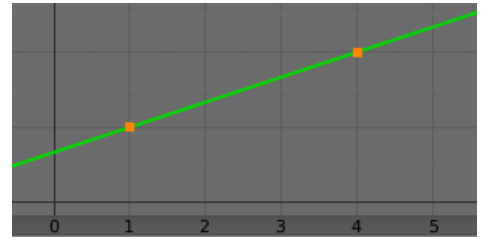
By default, the input and output values are clamped to the *Min/Max* values. When you enable *Extrapolate*, they’re allowed to go beyond these limits. This is illustrated with the graphs below, where the X axis represents the input (*Map From* set to *Min* = 1 and *Max* = 4) and the Y axis represents the output (*Map To* set to *Min* = 1 and *Max* = 2).

The Extrapolate option.





Extrapolate disabled: the output values are limited to the Map To range.



Extrapolate enabled: the output values can extend beyond the limits.

Target/Owner

Standard conversion between spaces. See [common constraint properties](#) for more information.

Influence

Controls the percentage of effect the constraint has on the object. See [common constraint properties](#) for more information.

Map From

The transformation to read from the *Target* (or *Bone*).

Location, Rotation, Scale

The type of transformation to read.

Mode Rotation

The type of rotation to use, including different [Euler](#) orders, [Quaternion](#), and other [Rotation Channel Modes](#). Defaults to using the [Euler](#) order of the constraint owner.

In the *Quaternion* mode, the channels are converted to weighted angles in the same way as the swing angles of the [Swing and X/Y/Z Twist](#) mode

X/Y/Z Min, Max

The input value range for each axis.

Map To

The transformation to apply to the owner.

Location, Rotation, Scale

The type of transformation to apply.

Order Rotation

Which [Euler](#) order to use. Defaults to the order of the constraint owner.

X/Y/Z Source Axis

For each of the three output axes, lets you choose the input axis that it should take its value from. You can select the same input axis multiple times.

Min, Max

The output value range for each axis.

Mix

Specifies how the result of the constraint is combined with the existing transformation. The set of available choices varies based on the type of transformation.

Replace

The result of the constraint replaces the existing transformation.

Multiply Scale

The new values are multiplied with the existing axis values.

Add Location Rotation

The new values are added to the existing axis values.

Before Original Rotation

The new rotation is added before the existing rotation, as if it were applied to a parent of the constraint owner.

After Original Rotation

The new rotation is added after the existing rotation, as if it were applied to a child of the constraint owner.

Note

- For historical reasons, the *Mix* mode defaults to *Add* for location and rotation, and *Replace* for scale.
- When using the rotation of the target as input, whatever the real values are, the constraint will always “take them back” into the (-180 to 180) range. E.g. if the target has a rotation of 420 degrees around its X axis, the values used as *X* input by the constraint will be:

$$\backslash(((420 + 180) \bmod 360) - 180 = 60 - 180 = -120\backslash)$$

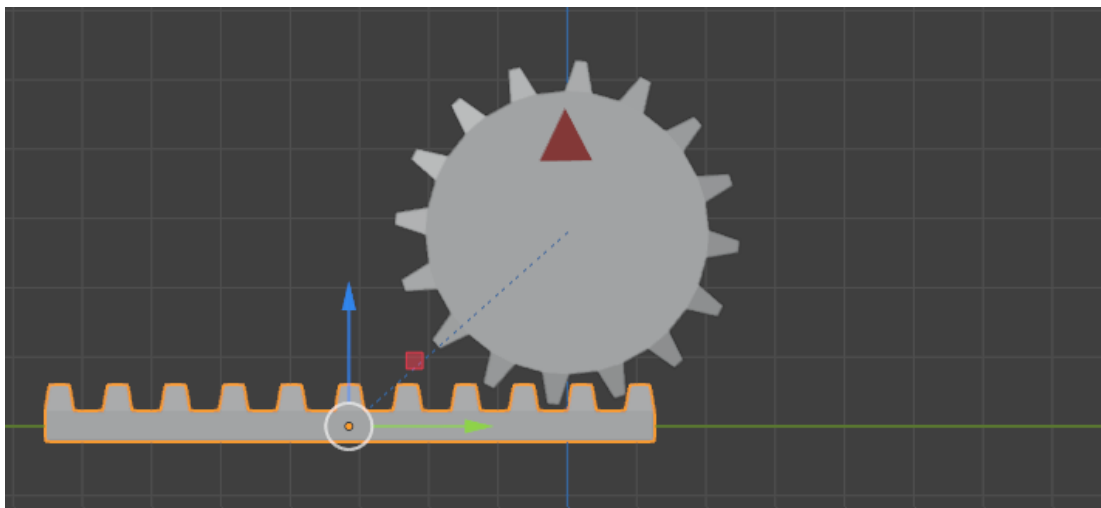
As such, this constraint is not really suited for transforming an object based on a gear’s rotation. Rotating a gear based on an object’s transformation works fine, however.

- Similarly, when using the scale transform properties of the target as input, whatever the real values are, the constraint will always take their absolute values (i.e. invert negative ones).
- When a *Min* value is higher than its corresponding *Max* one, both are considered equal to the *Max* one. This means you cannot create “reversed” mappings.

Example

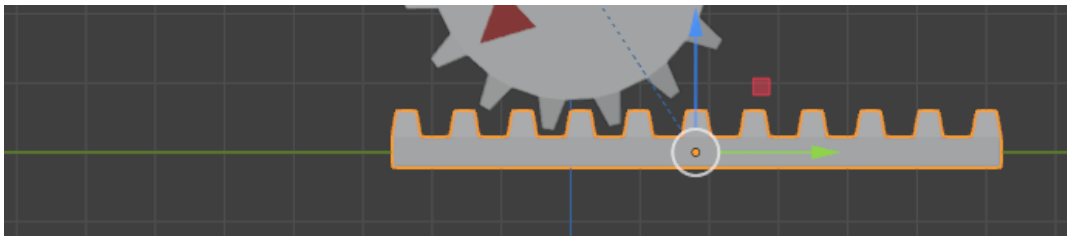
In the following example, we add a constraint to a gear that sets its X rotation based on the Y position of a rail:

- Target: Rail object
- Map From: Location
 - Y Min: -3m
 - Y Max: 3m
- Map To: Rotation
 - X Source Axis: Y
 - X Min: 0°
 - X Max: 180°



Before moving the rail.





After moving the rail.

By default, the gear will stop rotating if the rail goes outside the $(-3\text{m}, 3\text{m})$ range. You can enable *Extrapolate* to change this.

[Previous](#)
[Maintain Volume Constraint](#)

[Copyright](#) © : This page is licensed under a CC-BY-SA 4.0 Int. License

Made with [Furo](#)

Last updated on 2025-05-10

[No](#)
[Transform Cache Constr](#)

[View Source](#)
[View Translation](#)
[Report issue on this page](#)