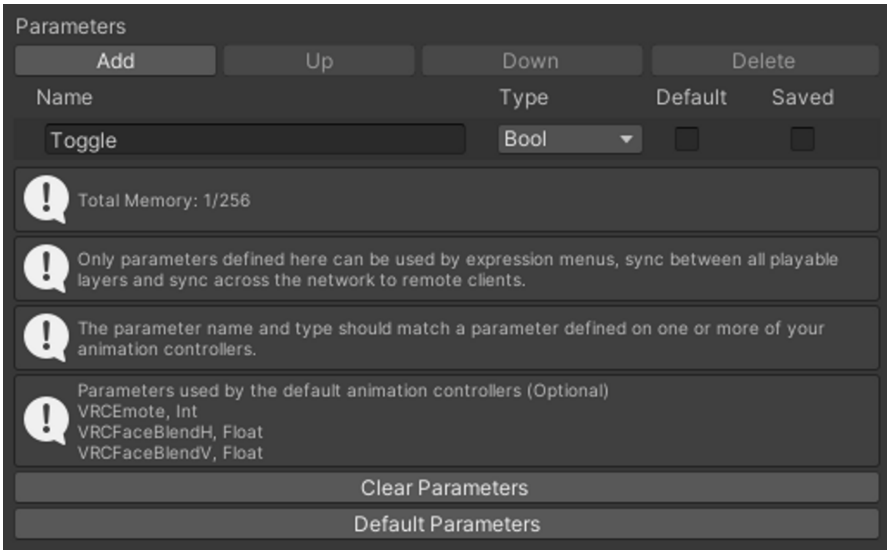
Expression Parameter Mismatch

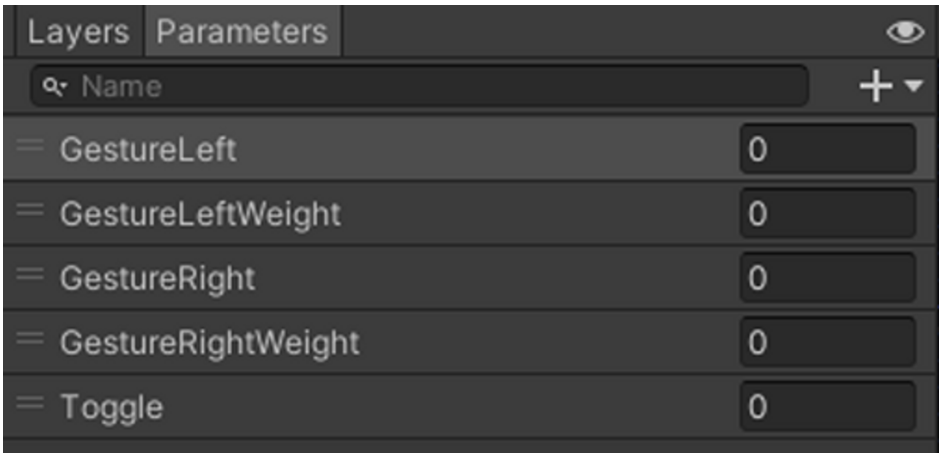
Original page: https://notes.sleightly.dev/parameter-mismatching/

In typical VRChat Avatars 3.0 creation, VRChat expression parameters are expected to be set as the same parameter type (`Bool`, `Int`, `Float`) as their respective local animator parameter counterparts.

However, parameters between your Expression Parameters vs your local animator parameters can have mismatched parameter types and still function. This behaviour differs from type to type, and is documented below.



A parameter labeled `Toggle` added to VRChat Expression Parameters as a `Bool`



A parameter labeled `Toggle` added to local animator parameters as a `Float`

Verified Parameter Mismatch Behaviour:

Expression Parameter Bool:

Animator Controller Parameter	Expression Bool = False	Expression Bool = True
Bool → Bool	Bool = False	Bool = True
Bool → Int	Int = 0	Int = 1
Bool → Float	Float = 0.0	Float = 1.0

Expression Parameter Int:

Animator Controller Parameter	Behaviour
Int → Bool	Any Int value above 0 sets bool to True
Int → Int	Expected Behaviour
Int → Float	Straight Conversion: Ex: Int = 2 → Float = 2.0

Expression Parameter Float:

Animator Controller Parameter	Behaviour
Float → Bool	Any Float value above 0 sets bool to True
Float → Int	Rounded Conversion: ≥0.5 → 1 : <0.5 → 0
Float → Float	Expected Behaviour

Why does mismatching work?

Unity Animators use floats on the C# back-end regardless of parameter type, while VRChat uses SBytes instead of bools, ints, or floats.

The Animator window and VRCSDK themselves makes the user experience simpler by allowing you to select the parameter types from the interface.

In this sense, the parameters are not being casted but **mismatched**.

Mismatching parameters also works with the VRChat native parameters.

Both the AV3 Emulator and the GestureManager have implemented Parameter Mismatch at this point in time

---

Revision #3
Created 2023-09-04 17:39:52 UTC by jellejurre
Updated 2023-09-04 17:46:24 UTC by jellejurre