

CVR Parameter Stream

This component allows you to use a multitude of local variables as inputs for Animators, Variable Buffers, Avatar Parameters, and Spawnable parameters.

As of writing, this component is only officially supported on Avatars to set Avatar Parameters, but does function as intended on Spawnables and in Worlds.

This script is underdeveloped in its current state and likely will need a full rework in the future. It is incredibly powerful and functions, but note that some aspects of this script are easily abusable or not fully thought out.

Note on Update Rate

The script will update entries every 0.05s for performance reasons, so depending on your usecase you may notice values are choppy and need smoothing via Animator shinanigans.

Parameter Name

The string of the parameter you want to apply the value to.

Target

Depending on the selected **TargetType**, this GameObject is used to get the Animator or Variable Buffer components to set the outputted value to.

Type

The local variable used to drive the selected Animator Parameter.

Some Type entries are not fully developed, non-functional, or poorly implemented.

- TimeSeconds
 - DateTime.Now returned as Float.
- TimeSecondsUtc
 - DateTime.UtcNow returned as Float.
- DeviceMode
 - If in VR returns 1, otherwise returns 0.

- HeadsetOnHead
 - If in VR & Headset is on head returns 1, otherwise returns 0.
This does not function at the time of writing.
- ZoomFactor
 - Returns the current Desktop Camera Zoom value between 0 and 1.
This value should be direct 0-1 based on if not zoomed or fully zoomed.
- ZoomFactorCurve
 - Returns the current Desktop Camera Zoom Curve value between 0 and 1.
This value should be smoother than the above type when zooming.
- EyeMovementLeftX
 - Returns eyeAngle.x from CVREyeController.
This is currently bugged & poorly implemented.
- EyeMovementLeftY
 - Returns eyeAngle.x from CVREyeController.
This is currently bugged & poorly implemented.
- EyeMovementRightX
 - Returns eyeAngle.x from CVREyeController.
This is currently bugged & poorly implemented.
- EyeMovementRightY
 - Returns eyeAngle.x from CVREyeController.
This is currently bugged & poorly implemented.
- EyeBlinkingLeft
 - Returns blinkProgress from CVREyeController.
This returns the total blinking progress and is not independent of the other eye.
- EyeBlinkingRight
 - Returns blinkProgress from CVREyeController.
This returns the total blinking progress and is not independent of the other eye.
- VisemeLevel
 - Returns visemeLoudness from CVRVisemeController.
VisemeLevel is constantly adjusting to your volume level. Consistent voice volume will be pretty accurate, with visemeLoudness being scaled to your average maximum volume. Sudden loud noises will cause VisemeLevel to become less sensitive until it readjusts to your average volume.
- TimeSinceHeadsetRemoved
 - Returns the amount of time you've had your headset off for.
This does not function at the time of writing.
- TimeSinceLocalAvatarLoaded
 - Returns the time since you've last switched into an avatar.
- LocalWorldDownloadPercentage
 - Returns the current normalized download percentage while loading a World.
Note that once the world has finished downloading & started loading the value will return to 0.
- LocalFPS
 - Returns your current FPS as a float.

- LocalPing
 - Returns your current Ping as a float.
- LocalPlayerCount
 - Returns the current Player Count in the World as a float.
Can never hit 0 as it always adds 1 to account for the local player.
- LocalTimeSinceFirstWorldJoin
 - Time since you've loaded into your Home World after launching the game.
Can be used to read how long you've been in-game.
- LocalTimeSinceWorldJoin
 - Time since you've loaded into any World.
Can be used to read how long you've been in a World.
- LocalPlayerMuted
 - Returns 1 if Muted, 0 if Unmuted.
- LocalPlayerHudEnabled
 - Returns 1 if Hud is enabled, and 0 if disabled.
- LocalPlayerNameplatesEnabled
 - Returns 1 if Nameplates are enabled, and 0 if disabled.
Will still return 1 if Nameplates are set to Menu Only.
- LocalPlayerHeight
 - Returns the Player Height setting in General Settings as a float.
This value doesn't return exact Player Height. Internally this returns ($(\text{int})\text{PlayerHeight} / 100f / 1.084f$).
You will likely need to play around with this value to tune it for your usecase.
- LocalPlayerControllerType
 - **Not implemented.**
- LocalPlayerFullBodyEnabled
 - Returns 1 if FBT, 0 if otherwise.
This returns 1 if Full Body Tracking is AVAILABLE, not CALIBRATED.
- TriggerLeftValue
 - Returns the Left Trigger value between 0 and 1.
- TriggerRightValue
 - Returns the Right Trigger value between 0 and 1.
- GripLeftValue
 - Returns the Left Grip value between 0 and 1.
- GripRightValue
 - Returns the Right Grip value between 0 and 1.
- GrippableObjectLeft
 - **Not implemented.**
- GrippableObjectRight
 - **Not implemented.**
- TransformGlobalPositionX
 - **Not implemented.**
- TransformGlobalPositionY
 - **Not implemented.**
- TransformGlobalPositionZ
 - **Not implemented.**

- TransformGlobalRotationX
 - **Not implemented.**
- TransformGlobalRotationY
 - **Not implemented.**
- TransformGlobalRotationZ
 - **Not implemented.**
- TransformLocalPositionX
 - **Not implemented.**
- TransformLocalPositionY
 - **Not implemented.**
- TransformLocalPositionZ
 - **Not implemented.**
- TransformLocalRotationX
 - **Not implemented.**
- TransformLocalRotationY
 - **Not implemented.**

TargetType

The target you want to use the output for.

Avatar Animator is currently the only supported & selectable TargetType as defined in CCK3.4, but internally CVRParameterStream supports these other TargetTypes and are currently functional in-game.

- Animator
 - Uses **Target** GameObject property to get Animator component.
 - Output will always be a Float.
- Variable Buffer
 - Uses **Target** GameObject property to get Variable Buffer component.
 - Output will always be a Float.
- Avatar Animator
 - Sets parameters onto the **Local Player Avatar**.
 - Output will be cast from Float to the correct Parameter Type.
Anything above 0.5f will return true if casting to Bool or Trigger.
- Custom Float
 - Sets parameters onto the found **Spawnable** component.
 - Output will always be a Float.

CVRParameterStream runs locally for every player, so keep in mind Network Sync when utilizing on Spawnables via the Custom Float TargetType. It may be best to use the Animator TargetType instead to avoid the Spawnable's ownership constantly changing as every client attempts to sync parameters to the parent Spawnable.

Nothing prevents you from using Avatar Animator TargetType on Spawnables or Worlds, but the game will now refuse to set parameters if the target CVRAvatar is not your local CVRAvatar component. (this was abusable for a period of time)

Static Value

A float value used to adjust values based on the selected **Application Type**.

Application Type

Allows you to choose from a list of preset arithmetic to adjust the returned values. For explanation sake, we will define a few variables:

StaticValue = the **Static Value** Float set above.

CurrentValue = the current **Avatar Animator** parameter value as a Float.
(This is always 0 for all other TargetTypes.)

ReturnedValue = the returned value from the above **Type** entry.

FinalValue = the final value after the **Application Type** arithmetic.

- Override
 - Sets returned value directly with no adjustment.
FinalValue = ReturnedValue;
- AddToCurrent
 - Adds returned value to current animator value.
FinalValue = CurrentValue + ReturnedValue;
- AddToStatic
 - Adds static value to returned value.
FinalValue = StaticValue + ReturnedValue;
- SubtractFromCurrent
 - Subtracts returned value from current animator value.
◦ **FinalValue = ReturnedValue - CurrentValue;**
- SubtractFromStatic
 - Subtracts returned value from static value.
◦ **FinalValue = StaticValue - ReturnedValue;**
- SubtractWithCurrent
 - Subtracts returned value from current animator value.
◦ **FinalValue = CurrentValue - ReturnedValue;**
- SubtractWithStatic
 - Subtracts static value from returned value.
◦ **FinalValue = ReturnedValue - StaticValue;**
- MultiplyWithCurrent
 - Multiplies the returned value with current value.
◦ **FinalValue = ReturnedValue * CurrentValue;**
- MultiplyWithStatic
 - Multiplies returned value with static value.

- **FinalValue = ReturnedValue * StaticValue;**
 - CompareLessThen
 - Returns 1 if returned value is less than static value, 0 if otherwise.
 - **FinalValue = ((ReturnedValue < StaticValue) ? 1f : 0f);**
 - CompareLessThenEquals
 - Returns 1 if returned value is less than or equal to static value, 0 if otherwise.
 - **FinalValue = ((ReturnedValue <= StaticValue) ? 1f : 0f);**
 - CompareEquals
 - Returns 1 if returned value is equal with static value, 0 if otherwise.
 - **FinalValue = ((ReturnedValue == StaticValue) ? 1f : 0f);**
 - CompareMoreThenEquals
 - Returns 1 if returned value is more than or equal to static value, 0 if otherwise.
 - **FinalValue = ((ReturnedValue >= StaticValue) ? 1f : 0f);**
 - CompareMoreThen
 - Returns 1 if returned value is more than static value, 0 if otherwise.
 - **FinalValue = ((ReturnedValue > StaticValue) ? 1f : 0f);**
 - Mod
 - **FinalValue = ReturnedValue % Mathf.Max(Mathf.Abs(StaticValue), 0.0001f);**
 - Pow
 - **FinalValue = Mathf.Pow(ReturnedValue , StaticValue);**
-

Revision #5

Created 5 December 2022 01:07:48 by NotAKidoS

Updated 23 December 2022 07:09:15 by NotAKidoS