Debugging Tracking Issues

SteamVR™ Tracking

Introduction

A variety of errors can contribute to a well-designed trackable object not “booting” into the VR space. This document serves as a guide to help diagnose the most common tracking issues and get you up and running as quickly as possible. If you are having trouble with your device, please refer to the section that best describes your issue and follow the guide there.

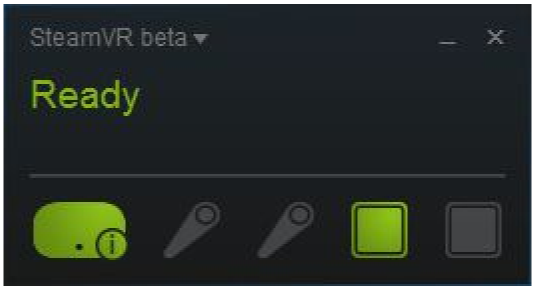
Note: SteamVR only downloads the JSON file from a connected device once per session. As you are debugging, you may change the JSON file on your device. SteamVR will not acknowledge this updated JSON until SteamVR is closed and reopened. As you follow these debugging steps, be sure to restart SteamVR after every modification of the JSON file.

My Object Does Not Appear in SteamVR

There are a number of completely different issues that can cause an object to not appear at all in SteamVR. Please narrow down your symptoms in the following sections.

My object’s controller icon does not show up in the SteamVR window

Normally, connecting a controller to SteamVR produces a flashing green controller icon that eventually turns solid green once tracking has been established. If you do not see at least a flashing green controller icon, try the following:



Ready

Verify that your controller’s hardware is functional, turned on, and connected/paired

Try connecting to the controller using `lighthouse_console` You should see your controller’s (or your paired dongle’s) serial number after entering the `serial` command. Verify that you are connected to the correct controller by using the `imu` and `dump` commands and shaking the controller. If you are connected and your controller is functional, you should see the streaming IMU data react to your movements.

If your controller does not show up in `lighthouse_console`and/or does not respond to the `imu` command, there may be something wrong with your controller. Ensure that it is powered on and connected and that it is paired properly.

Check for formatting errors in your JSON file

If there are any anomalies in your controller’s JSON file that render it an invalid JSON file, SteamVR will not know what kind of device it is and therefore will not show a controller icon in the SteamVR window. To verify, look at the log file at `C:\Program Files (x86)\Steam\logs\vrserver.txt`. Look for lines in the log file that mention your device’s serial number. Here’s an example where the controller’s JSON file had an extra comma:

```
<date> - lighthouse: LHR-C2544EA4: Read config of 1936 bytes from [vid:28de, pid:2300]
(LHR-C2544EA4) and inflated to 6120 bytes
<date> - lighthouse: HID opened: VID 28de PID 2300 serial LHR-C2544EA4 seq 1 | if 0
<date> - lighthouse: Can't parse HMD config from USB: * Line 19, Column 4
  18:       "position" : [ -0.029960000887513161, 0.0059998901560902596, -0.049249999225139618
],
  19:    },
  20:    "lighthouse_config" : {

  Missing '}' or object member name
```

Correcting the JSON file should resolve the issue in this case.

There is also a possibility that a firmware bug is preventing proper transmission of your JSON file from your controller or that your controller has no JSON file. Use `lighthouse_console` and verify that the `downloadconfig` command downloads the JSON file you intend to use. If `downloadconfig` fails, you may be running an old version of firmware. Follow the instructions on the SteamVR Tracking HDK forum to update your firmware to the latest build.

Verify that your JSON file includes the “type” member

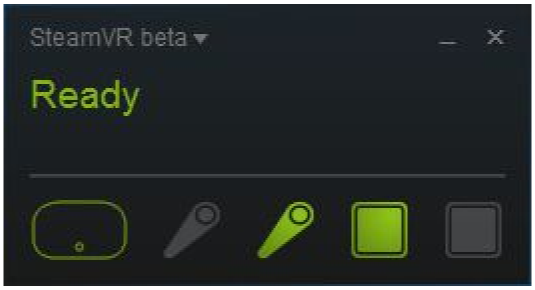
SteamVR will only honor an object’s JSON file if contains the member `“type”` set to the value `“Lighthouse_HMD”`. `Lighthouse_HMD` is the required value even if your device is a controller.

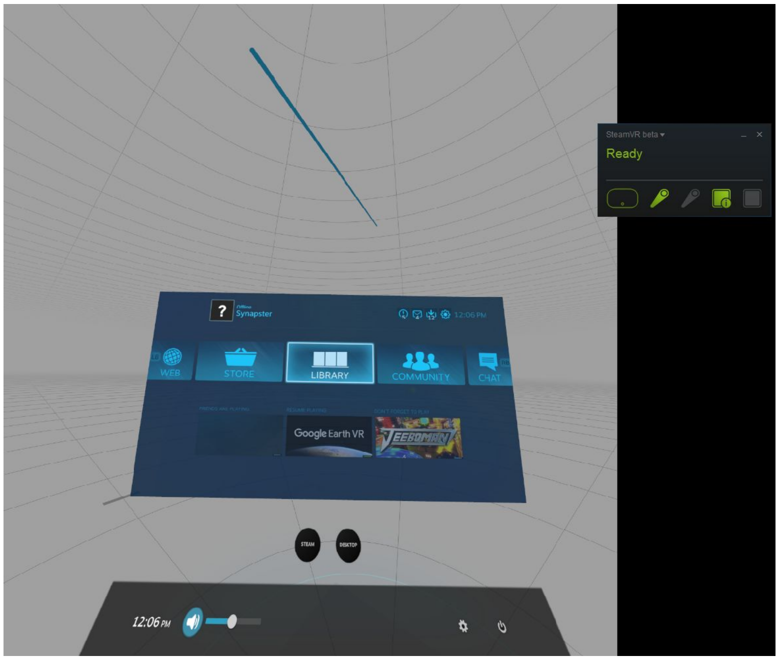
Verify that this text is somewhere in your object’s JSON file:

```
"type" : "Lighthouse_HMD"
```

My object’s controller icon glows solid green, but I still can’t see it in VR

If your controller is shown as solid green but does not appear in SteamVR’s window, the first thing you want to do is establish if it is tracking, but simply not visible. If your system button is still accessible, try pushing it to load the SteamVR menu. If your controller is tracking properly, you should see a laser pointer emanating from your controller’s location as shown below.





If you cannot access the system button, you can also check in the file `C:\Program Files (x86)\Steam\logs\vrclient_vrcompositor.txt` for relevant errors like those listed below:

```
<date> - Render model <rendermodels>\ref_controller\ref_controller.obj has 0
texture coordinates, expected 82994
```

```
<date> - Render model <rendermodels>\ref_controller\ref_controller.obj has 0
texture coordinates, expected 82994
```

```
<date> - Render model from <rendermodels>\ref_controller\ref_controller.obj
contained more than one shape. We only support one.
```

```
<date> - Unable to load image C:\Program Files
(x86)\Steam\steamapps\common\SteamVR\resources\rendermodels\ref_controller\C:\Us
ers\user_name\Desktop\ref_controller.png: can't fopen
```

If you see one of these or similar errors, try re-exporting your rendermodel and testing it in MeshLab. Typically, a rendermodel .obj file that works in MeshLab will also work in SteamVR.

If pressing your system button brings up the SteamVR menu, but you are not able to see the laser beam coming from your controller, double check your JSON file's modelNormals. A modelNormal vector that has a magnitude of zero will produce an error state where the controller's icon is solid green, but the controller is not tracking.

My object tracks well once booted, but takes a very long time to boot

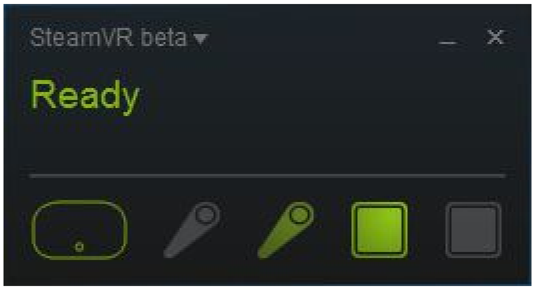
SteamVR is constantly running various calibration routines and heuristics on hardware after it is connected. It is possible that your device has poor sensor placement and it takes SteamVR a long time to account for this so it can begin tracking. This process is repeated every time the controller is connected.

If your controller takes a substantial amount of time to boot, try verifying your sensor placement using the visualize tool of `hmd_designer_gui`. A single misplaced sensor can cause a problem like this as it takes a while for SteamVR to learn to ignore that particular sensor. If your sensors are all placed correctly and your `channelMap` has no errors, try running the optical calibration routine to improve boot time.

SteamVR also prevents the object from booting until the IMU and optical sensors agree on the object's pose and motion. If the IMU has not been calibrated, or was calibrated poorly, the object may not boot immediately. SteamVR calibrates the IMU as the object tracks; so, over time, the calibration may correct itself enough to boot the object. If you object takes several seconds to boot but tracks well once booted, try recalibrating the IMU to eliminate the delay.

My object's icon will not stop flashing green in SteamVR

There are no specific causes that are unique to a flashing non-tracking controller and don't also apply to a poorly tracking controller. Please refer to the "My object tracks poorly" section for help.



My object tracks poorly

When your device has trouble tracking, it is recommended that you turn off one of your base stations and attempt to track with a single base station. This will help you narrow down issues and simplify debugging. With a single base station in operation, use the following sections to help resolve your issue.

My object only tracks when held stationary in certain orientations

Good tracking only when certain surfaces face the base station

If your object only tracks when certain regions of its surface are facing the base stations, there may be an issue with a small number of sensors on the opposite side.

You can often narrow down the problematic sensor by selectively covering sensors with IR opaque tape to see if that improves performance. Many times a misplaced sensor will negatively affect tracking more than an obscured sensor. If tracking improves when a particular sensor is covered, double check that that sensor is positioned and connected as specified in the JSON file. You can also use the visualize tool within `hmd_designer_gui` to confirm sensor placement and the channel map.

Good tracking when certain surfaces face the ground

If you find that your device will only track stably when held still with a certain surface facing the ground, you likely have an issue with your IMU alignment. With a poorly aligned IMU, the device may interpret the acceleration due to gravity as a lateral acceleration predicting a lateral motion that is at odds with the optical tracking system's measurements. If your IMU's actual orientation is only rotated around a single axis from its JSON programmed orientation, you may find that in some angles the gravity vector lines up with the unrotated axis and tracking is relatively stable.

Whenever you suspect the IMU is the problem, it’s best to disable it. You can disable the IMU for all tracking objects by closing SteamVR and then editing the file `C:\Program Files (x86)\Steam\config\steamvr.vrsettings`. If you have a brand-new install of SteamVR, you may not find this file. It will be generated the first time you change a setting in SteamVR.

Somewhere in the top level of the JSON format, add the following:

```
"driver_lighthouse" : {
   "disableimu" : true
}
```

Add a comma at the end if appropriate. Relaunch SteamVR. If you did it correctly, you’ll note that tracking on all of your devices (including the HMD) is less smooth. With the IMU disabled, your problematic device may start tracking better. Note that its motion will not be completely smooth as the `disableimu` flag intentionally increases gain in order to amplify minor tracking issues for diagnostics, but when held steady, your object should remain relatively motionless for all orientations.

If your device begins tracking better with the IMU disabled, use the visualize tool in `hmd_designer_gui` and verify that your IMU orientation is correct.

You can re-enable the IMU by setting `“disableimu”` to `false`, or deleting that line from `steamvr.vrsettings`.

My object only tracks well when held still or translated, but loses tracking when rotated quickly

Rotation-related tracking issues are usually the result of poor IMU positioning. Assuming orientation is correct, the IMU will see the correct accelerations when the object is held still or translated. When rotated however, the IMU registers a rotation, but the acceleration anticipated from the IMU’s programmed location does not align with the actual acceleration it experiences.

To diagnose this problem, try disabling the IMU using the guide in the previous section and see if tracking performance improves. You can also use the visualize tool to verify IMU placement.

My object only tracks well with the IMU disabled, but its orientation and location are correct

If you’ve verified your IMU’s location and orientation using the visualize tool, you may have a problem with your IMU’s calibration. Open your device’s JSON file and look at the IMU section. Your calibration data should look something like this:

```
"acc_bias" : [ 0, 0, 0 ],
"acc_scale" : [ 1, 1, 1 ],
"gyro_bias" : [ 0, 0, 0 ],
"gyro_scale" : [ 1, 1, 1 ],
```

With values that are close to either 0 or 1. If you have values that differ substantially from 0 or 1, you may need to run `imu_calibrator` again. `imu_calibrator` works best when used with some sort of apparatus that can guarantee that your object is being held in orientations that are as orthogonal as possible. If you have been performing calibration by holding the device in your hand, it may improve your results to use a jig.

Also note that you should only use a calibration if your accelerometer fit error is less than 0.1. If you get an error greater than 0.1, try running calibration again.

My bias and scale are very large no matter how I calibrate

If your calibration results in `acc_bias`values as high as 10-15, you may have a defective IMU. To confirm IMU functionality, open `lighthouse_console`and use the `dump` and `imu` functions. If your device is held still, you should see `gyro` values close to 0 and `accel` values that are not much higher magnitude than 9.8.

If you see very large `gyro` and `accel` values in certain orientations when the device is held still, you may have a damaged or defective IMU. Try replacing the IMU on your PCB. Also keep in mind that ultrasonic welding and ultrasonic cleaning processes can damage IMUs. If you are applying an ultrasonic process after soldering the IMU to your PCB, you may want to try adjusting that process to add the IMU later or build some better shock absorbtion into your design.

My render model does not line up with my device

The rendermodel appears very close to the base station or very far away no matter where the actual device is

If your JSON file's `modelPoints`field is in the wrong units, it's possible that it will report your controller's size as being much larger or much smaller than actual. SteamVR will attempt to reconcile this by placing the controller very far away or very close to the single base station.

Make sure that your modelPoints values are specified in meters.

The render model is rotated from my actual device

The `head` section of your JSON file determines the orientation and location of the origin of your rendermodel relative to the `modelPoints`coordinate system. Verify that the `plus_x` and `plus_z` variables indicate the direction of the render model's +X and +Z axes in the `modelPoints`system.

Remember that the "laser pointer" always points in the -Z direction of your render model, but the render model can be rotated around the physical model with `head`. If your laser is not pointing out of the correct side of your rendermodel, use your 3D art program to rotate and re-export the Wavefront OBJ file.

The render model is translated from my actual device

The `position`section of the `head`variable will determine the translational offset of your device's render model relative to the `modelPoints`axes. Verify that your offset is set correctly.

If your `position`is correct and your device still has an offset, you may have a persistent translation error due to a lack of baseline in your sensors. Verify that all of your sensors are functioning using the `sample` and `dump` commands in `lighthouse_console` If all sensors are working but the problem persists, use the simulation tools within `hmd_designer_gui`to evaluate your device's translation error. You may need to try a different sensor placement strategy or add more sensors.

My object will only track or calibrate in certain locations or at certain times of day

Early prototype devices rarely have adequate sensor covering. If you work in an environment that has a large amount of visible light, you may be saturating your sensors which are partially sensitive to shorter wavelength visible light. Common sources of interfering light are fluorescent lights or the sun.

Try moving to a new area with less light to see if that improves performance and then plan to re-evaluate performance once you have developed your sensor covering with visible light filters.

It is also possible that sources of hot or cold air may be diffusing the laser light coming from the base stations. Look for hot appliances or heating/cooling vents in the vicinity of your base station.

Individually, my devices track fine, but when connected to the same computer, some of them fail to show up in SteamVR

SteamVR currently allows for up to 16 tracked objects which includes the HMD, the two base stations, and 13 controllers. Any devices connected after these 16 will not work.

If you are still having trouble with fewer than 13 controllers, try using different USB ports or hubs with higher power capabilities.

My object will not calibrate

I don’t see any “dots”

Sometimes you may load `vrtrackingcalib` and find that your calibration gets stuck at this step:

```
CDisambiguatorFramer: Non monotonicity in ProcessChannelHit

Last sample 6 6497900795.000000 6497905836.000000

New sample  6 6421444603.000000 6421449295.000001

base 00000000 u2 r2 (i1)
```

In order for `vrtrackingcalib` to calibrate your device, it must first have enough information to establish an initial position. If your sensor locations vary dramatically from what your JSON file describes, vrtrackingcalib will be unable to start calibration. Use the visualization tool of `hmd_designer_gui` and double check your sensor positions and channel map assignments.

It’s also possible that your base station is not switched on or you have more than one base station present. Make sure you have a single base station and it is producing sync pulses and laser sweeps before calibration.

Calibration won’t finish, and some of the sensor numbers say “(0-hits)”

The second number provided to `vrtrackingcalib`’s `/bodycal` flag specifies the minimum number of hits for each sensor input listed in the JSON file’s channelMap. If you have a sensor channel listed, and that channel is not receiving any optical data, `vrtrackingcalib` will indicate this with `(0-hits)`.

Try adjusting your grip to point the affected sensor towards the base station. You can use `hmd_designer_gui`’s visualize tool to determine the location of the sensor.

If adjusting your grip does not help, use `lighthouse_console`’s `dis` and `period` commands to determine if your affected sensor is receiving any hits from the base station. A properly configured JSON should have a matching list of channels in the `channelMap` and the output of the `period` command. It’s possible that all of your sensors are transmitting data, but your JSON may be looking for data in a disconnected port and ignoring data in a connected port that is not listed in the `channelMap`.

If you still aren’t seeing any data from a particular sensor, double-check its connection. The sensor ribbon cable should be oriented with the metal contacts facing up, away from the PCB on both ends of the ribbon. Also inspect the ASIC on the sensor for damage.

None of the sensors say “(0-hits),” but calibration still won’t finish

(0-hits) will go away as soon as the associated sensor gets a single hit. It’s possible that your controller has gotten at least one-hit to each sensor, but the way you are holding it is preventing some sensors from getting the minimum number of hits as specified by the second argument to the `/bodycal` flag of `vrtrackingcalib`.

Try repositioning your grip to expose the affected sensors to the base station. You can verify their location on your device using the visualize tool of `hmd_designer_gui`. Note that a sensor with very few hits compared to the other sensors will take a long time to disappear once exposed to the base station, since it is basically starting from zero.

It’s also possible that one of your sensors is experiencing a hardware issue where only some sensor hits are coming through. The `dis` and `period` commands of lighthouse_console should show a relatively even number of hits for all sensors in close proximity. If one sensor is receiving substantially fewer hits, it is likely to be a hardware issue. Inspect the affected sensor for a damaged ASIC or a faulty ribbon cable connection.

Finally, JSON files that are mostly correct but have a small number of sensors substantially out of place will take a long time to finish calibration, and the resulting calibration file will not work should calibration actually complete. Double-check your sensor assignments and locations by visualizing the JSON file using `hmd_designer_gui`and try calibration again. After calibration, it can be useful to visualize the calibrated JSON file using `hmd_designer_gui`to verify that the sensors were only adjusted slightly.

I keep getting a WatchmanFIFOoverflow error while calibrating

The WatchmanFIFOoverflow error is usually a result of faulty data coming from a sensor channel. If your device is not properly insulated from the operator (you have exposed sensors/hardware), then it’s possible that your hands are introducing small currents that the system is interpreting as sensor hits. Adjust how you are holding the device and be careful not to touch any of the circuitry with your hands. Keep trying calibration until it completes.