

# VRPN server Installation help



Am Öferl 6  
D-82362 Weilheim i. OB  
Germany  
T +49 (0)881-92530-00  
F +49 (0)881-92530-01  
[www.ar-tracking.de](http://www.ar-tracking.de)

## 1. VRPN Trivia

VRPN is a daemon, which runs continuously on a computer to connect tracking hardware to software in a standardized way. The VRPN daemon receives data from DTrack by UDP packages. Applications can then open network connections to the VRPN daemon and retrieve the data from the daemon in a hardware-independent way. The VRPN daemon can also combine data from different sources and present it to the application in a consistent way. VRPN delivers the data from the ART devices in the following data types:

- Tracker: 6DOF pose (position and orientation) → For standard body and Flysticks
- Button: Reports events (press/release) from Flystick buttons
- Analog: Joystick information from Flysticks (only transmitted if value ≠ 0)

The Flystick1 device had a digital joystick, so only -1/0/1 in X and Y direction. For a pseudoanalog behavior, the analog value is increased in a ramp from 0 to 1 or -1 when pushing the analog stick into any direction. The duration of the ramp can be given in the configuration, typically 0.5s. Flystick2 and Flystick3 devices real analog joysticks and ignore this value. If the 'old data format' is activated in the Flystick configuration of DTrack2, Flystick2 and Flystick3 devices behave like Flystick1.

### Important: Numbering of Trackers

- VRPN maps standard bodies and Flysticks to 'Tracker' sensors, renumbering the Flysticks after the standard bodies. So if the config file defines 3 standard bodies, the standard bodies are reported with sensor IDs 0,1,2 and the first Flystick with sensor ID 3.
- The VRPN server can operate with a changing number of bodies automatically. Nevertheless, it is highly recommended to once define a sufficient number of standard bodies in the configuration file, even if not used in the current application, so the tracker IDs of the Flysticks do not change when adding new bodies.
- The DTrack display shows Body IDs starting from 1, while VRPN IDs start with 0. So DTrack's standard body #1 is VRPN's sensor #0, DTrack's #2 is VRPN's #1 and so on.
- All DTrack bodies are reported by the same device, e.g. "DTrack@localhost", and the body is being selected by the sensor ID.

### Warning: Non-ART device drivers removed

The VRPN server supplied on the A.R.T. web page has been stripped from all non-A.R.T. drivers, so it cannot receive any other information than DTrack data. If VRPN should be used to mix A.R.T. tracking data with 3rd party data, the public domain distribution must be downloaded from the Internet and used. By default, public domain VRPN servers contain DTrack support.

The installation of VR applications with VRPN is a two-step procedure:

1. Configure VRPN correctly so it receives the DTrack data
2. Configure the application so it reads its data from the VRPN daemon

### Startup shell scripts

The MS Windows VRPN server distributed by ART contains DOS/Windows batch files for starting the server and the test application:

- `startserver.bat` starts the VRPN server using the `vrpn.cfg` file in the same directory.
- `starttest.bat` starts the VRPN test program, assuming that the VRPN server name is DTrack and assuming that the server is running on the same computer (`localhost`)

These scripts rely on the executables being in the same directory as script.

## 2. Installation of the DTrack Server

With a current DTrack version, your vrpn.cfg file should include a line like the following:

```
vrpn_Tracker_DTrack DTrack 5000 0.5 6 1
```

With this line, you receive the data from DTrack on port 5000 (ART standard port) and serve it to the application under the name 'DTrack' with a maximum of 6 standard bodies and 1 Flystick. The 0.5 value is the 'ramp' time for the old Flystick1 and ignored when no Flystick1 devices are used.

### **WARNING: names in VRPN are case-sensitive**

If the number of targets is not specified, our data output must send 6dcal data, which is not active by default.

Then you start a command shell and start the VRPN server. It is recommended to use absolute paths, both in Linux and Windows, if the vrpn config file does not reside in the same directory as the server executable. By default, the driver will look for a file named vrpn.cfg in the same directory as the executable.

If you use a different file (not recommended), you have to specify it with an option “-f <filename>”. If you do that, put the file in the same directory with the server or use absolute paths.

```
D:\vrpn>vrpn_server.exe -v -millisleep 0
Reading from config file D:\vrpn\vrpn.cfg
Opening vrpn_Tracker_DTrack: DTrack at port 5000, timeToReachJoy 0.00
```

The 'millisleep' option is important: without it, VRPN will create a high CPU load. The parameter can also be set to 1, in which case the CPU load is reduced even further, on the price of a slightly increased delay.

Now open a second shell to test whether the data is correctly received. Do not stop the server! There is a test application in the 'Utils' sub-directory of the VRPN directory:

```
D:\vrpn\Utils>.\vrpn_print_devices.exe -trackerstride 30 DTrack@localhost
Opened DTrack@localhost as: Tracker Button Analog Dial Text.
Press ^C to exit.
Tracker DTrack@localhost, sensor 0:
pos ( 0.80, 0.29, 0.87); quat ( 0.24, -0.22, 0.90, 0.31)
Tracker DTrack@localhost, sensor 1:
pos ( 0.15, -0.01, 0.94); quat (-0.07, -0.76, -0.01, 0.65)
Tracker DTrack@localhost, sensor 0:
pos ( 0.79, 0.28, 0.87); quat ( 0.24, -0.22, 0.90, 0.31)
Tracker DTrack@localhost, sensor 1:
pos ( 0.12, -0.01, 0.93); quat (-0.07, -0.76, -0.04, 0.65)
Tracker DTrack@localhost, sensor 0:
pos ( 0.77, 0.28, 0.87); quat ( 0.24, -0.22, 0.89, 0.31)
Tracker DTrack@localhost, sensor 1:
pos ( 0.08, -0.01, 0.91); quat (-0.07, -0.76, -0.07, 0.65)
...
```

As soon as you see your tracking data data here, you have configured the VRPN server correctly.

### **WARNING: Tracking data is only generated when tracking targets are in the tracked area**

## 3. Configuration for „stateful firewalls“

Frequently, computers cannot receive UDP packets from the controller without first initializing the connection by sending a UDP packet to the controller. To do that, the controller's IP number or name must be known. Then, the following syntax will send an initialization package to the controller for a controller at IP 1.2.3.4 or a controller named “controller.my.net” sending its data to port 5000 on the VRPN server computer:

```
vrpn_Tracker_DTrack DTrack 1.2.3.4:5000:fw 0.5 6 1
vrpn_Tracker_DTrack DTrack controller.my.net:5000:fw 0.5 6 1
```

## 4. Receiving data via Multicast

Multicast is a protocol on an IP network, which allows data packets to be sent to all computers in the network simultaneously. This is done by opening the network connection in MCAST mode and then sending the packet to a specified port at an address in the number range 224.0.1.0 – 239.255.255.255. Any computer who wants to receive the data opens an MCAST receiver with the same IP number and port and will then receive all packets sent. The advantage of this method is that many computers in the network can receive the same data without having to send it multiple times. This saves both latency and bandwidth in the network.

To receive MCAST packets sent out to IP address 225.1.2.3 port 5000, the syntax is

```
vrpn_Tracker_DTrack DTrack 225.1.2.3:5000 0.5 6 1
```

Please be aware:

- The firewall on the local computer must allow receiving of MCAST packages: The “start package” method is not possible for Multicast.
- Only one process on a computer can receive data from a single IP/port multicast channel at a time

## 5. Advanced options

The full configuration line of the DTrack VRPN server may have additional optional parameters:

e.g.  
vrpn\_Tracker\_DTrack DTrack <port> <ramp> <nBod> <nFly> {map} {3DOF} {debug}

```
vrpn_Tracker_DTrack DTrack 5000 0.5 4 2 2 3 4 5 0 1 3d -
```

Field	Example	Meaning
Port	5000	Receive data at port 5000
	15.10.2.1:5000:fw	Receive data at port 5000, send initializing packet to controller at IP 15.10.2.1
	225.1.2.3:5000	Receive MCAST data, Multicast-address 225.1.2.3:5000
ramp	0.5	On Flystick1: Increase value from 0 to 1 in 0.5s On Flystick2/2+/3: ignored
nBod	4	Use standard bodies with ID=0...3, start Flysticks with ID=4: max. 4 std. bodies
nFly	2	Use max. 2 Flysticks
map	2 3 4 5 0 1  (All numbers from 0 to nBod+nFly-1 in any order)	Map VRPN bodies 0,1,2,3,4,5 to 2,3,4,5,0,1 Results for 2 used bodies and 1 Flystick: ART body #1 → VRPN sensor #0 → output VRPN sensor 2 ART body #2 → VRPN sensor #1 → output VRPN sensor 3 ART Flystick #1 → VRPN sensor #4 → output VRPN sensor 0
3DOF	“3d”	Also report 3D measurements as VRPN 3D data: Additional sensors after last configured Flystick, so starting at ID=6 in example
-	“-“	Dump input data in VRPN server window

## 6. Configuring your application

This is the part we can only give little help - here the application experts must come in. The only part we can tell you is that your configuration file has to include a line that defines the tracker to be used. This must include the name DTrack@localhost if the VRPN runs on the local machine and DTrack@<hostname> if it is on another machine. Then you have find the place to to assign the correct sensor numbers to the tracked objects of the application.

**Remember: VRPN tracker numbers = DTrack body number – 1**