

VRPN Tracking Documentation

VRED Professional 2016 SR1

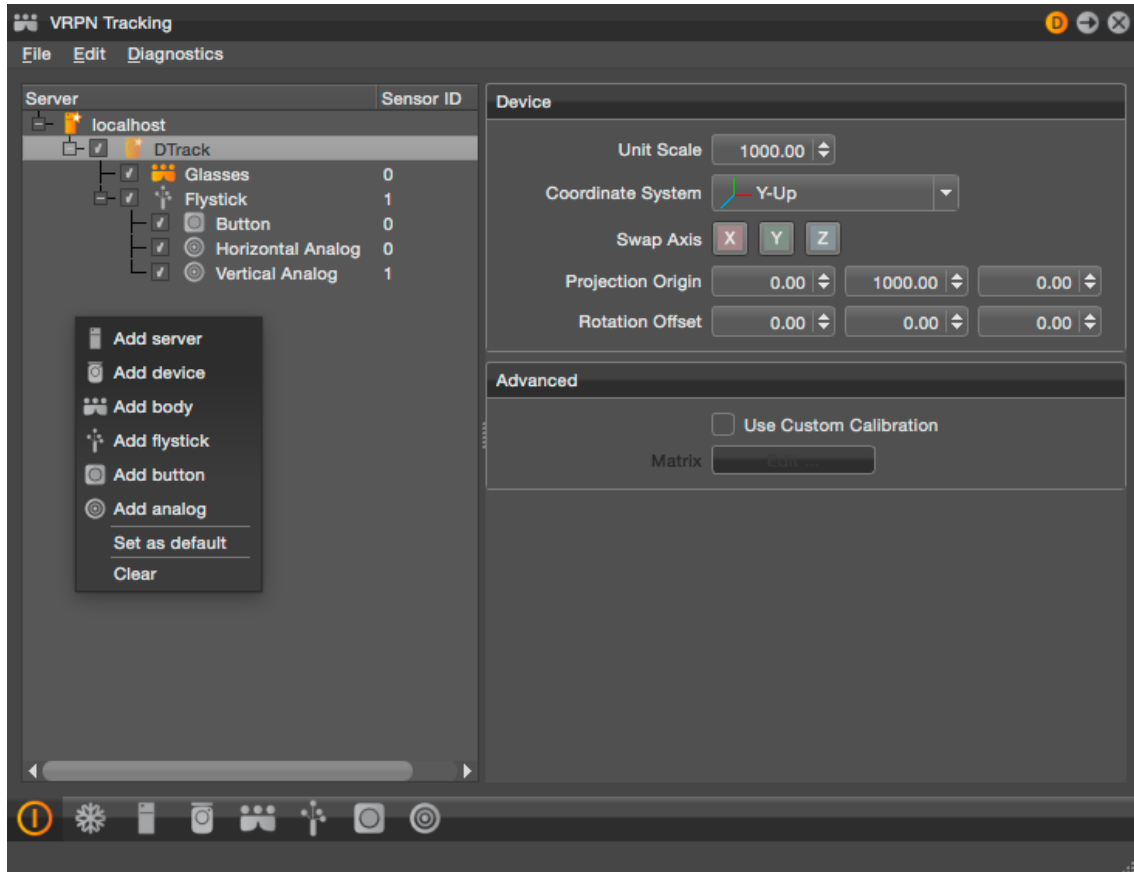
VRPN Tracking Documentation	1
Introduction	2
The user interface	2
Menu bar	2
Configuration list view	3
Attribute area	3
Quick-access bar	3
Configuration	4
VRPN server settings	4
Device settings	4
Body settings	5
Fly-stick settings	6
Advanced settings	7
Preferences	7
Python API	8
Examples	9
How to set up an A.R.T tracking configuration	9
Python script for button interaction	10
Trouble shooting	11
Nothing happens when tracking is enabled	11
VRPN server is not starting	11
Problems with the new user interface	11

Introduction

VRED Professional offers the possibility to receive data from various tracking systems via VRPN (Virtual-Reality Peripheral Network). Supported VRPN objects are trackers, buttons and analogs. This functionality allows you to set up power-wall and cave projections with head-tracking, object tracking, interactive navigation with fly-sticks and much more.

The module is located in the *Interaction* menu and is called *VRPN Tracking*.

The user interface



Menu bar

The menu bar holds actions to basic functionality of the module. It contains three menus: *File*, *Edit* and *Diagnostics*.

The file menu provides functions for clearing, loading and saving of tracking configurations. In the edit menu you can find actions to create new objects as well as a *Freeze* and *Reset Camera* option. The freeze option blocks the execution of incoming VRPN data while tracking is active. Reset camera will clear all set head projection and HMD values and puts the camera in the origin of the scene. The diagnostics menu has one entry, *Connection Test*, which opens a new dialog where the current configuration can be tested.

Configuration list view

The configuration list view shows a hierarchical representation of the tracking environment. It is divided into two columns: *Objects* and *Sensor ID*.

Every configuration exists of a general structure. Top-level items in the first column represent the VRPN servers. On the next level are the tracking devices. Each device again can have several children of the types body, fly-stick, button or analog. Bodies and fly-sticks can also have additional buttons and analogs. Furthermore a fly-stick has three predefined children (one button and two analogs). Orange coloured server and device icons indicate a valid connection to the server. An orange body or fly-stick icon shows that the sensor is receiving data. A star within the icon represents a default parent object.

The second column holds the identifiers for the sensor objects. Sensor objects are bodies, fly-sticks, buttons and analogs. The identifier order should match the one coming from the tracking application to get the desired behaviour.

A right mouse button click opens a pop-up menu with additional options. Those options include the creation of tracking objects as well as a *Set as default* and *Clear* option. *Set as default* is available for server and device items and specifies the object where newly created child objects are attached to if no parent item is selected. Choosing *Clear* will destroy the current configuration.

Attribute area

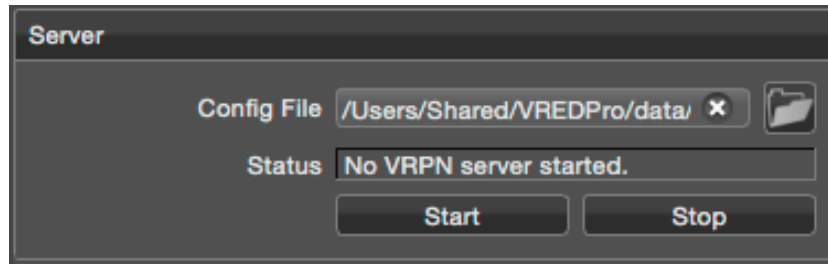
The attribute area displays the settings for the currently selected tracking object of the list view. It basically consists of one main settings frame and an optional advanced frame. Latter one is only displayed for selected device, body and fly-stick items. It provides legacy support for tracking configuration files of previous VRED versions.

Quick-access bar

The quick-access bar holds buttons for commonly used functions which are enable/disable tracking, freeze tracking and the creation of new objects.

Configuration

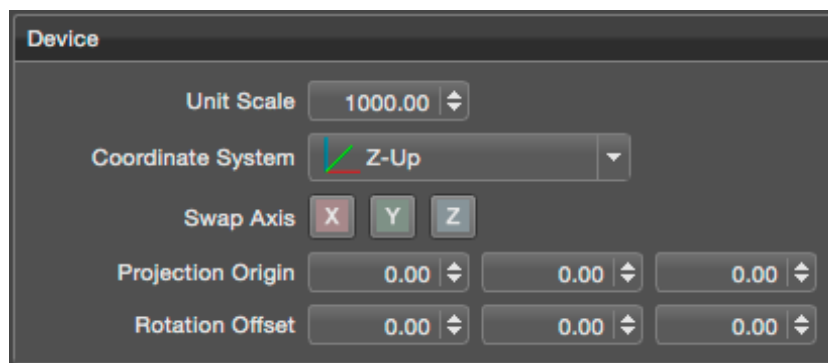
VRPN server settings



To receive data from a tracking system which supports VRPN, it is necessary to have a running VRPN server for transmission. If such a server is already running you simply have to specify the hostname or IP address. If no server is running you can start one within VRED. Therefore you have to specify a VRPN configuration file which is used when the server is started. Example configuration files are provided within the VRED installation package. In the status line shows whether starting was successful or not.

Config file	Specifies the path to a config file (*.cfg) which is used to start a VRPN server out of VRED. A right click offers the option to choose the default preference file which is specified in the preferences.
Status	Displays the current server status.
Start	Start a new VRPN server process.
Stop	Stop the running VRPN server.

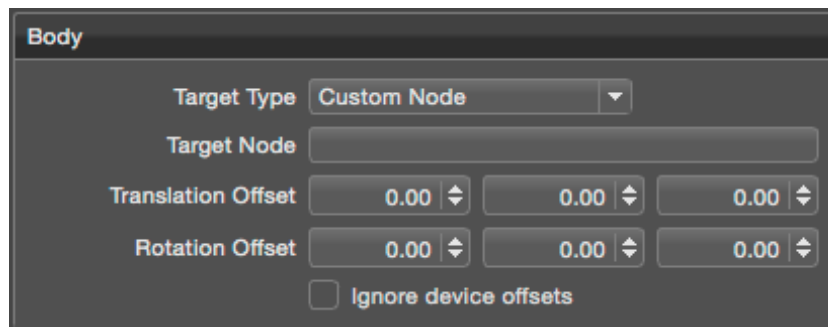
Device settings



A tracking device in VRED represents the physical tracking environment. The name has to correspond with the one used in the VRPN configuration. Further settings include the orientation of the tracking system as well as possible offset values.

Unit scale	Tracking values are normally transferred in metres. The unit scale converts the values to the VRED unit of millimetres. The default value is 1000. If the tracking system is not sending its values in metres it is necessary to adjust this value.
Coordinate system	There are three predefined tracking coordinate systems: <ul style="list-style-type: none"> • <i>X-Up</i>: X pointing up, Y left and Z front. • <i>Y-Up</i>: X pointing right, Y up and Z front. • <i>Z-Up</i>: X pointing right, Y back and Z up.
Swap axis	To match all kinds of coordinate systems it is additionally possible to swap one or more of the three predefined axes.
Projection origin	The projection origin specifies the position of the projection in global tracking coordinates. It is mainly used for head projections.
Rotation offset	Possibility to add an extra rotation.

Body settings



For every device it is possible to define one or more tracker bodies. A body basically represents a simple sensor in the tracking system. Its identifier can be specified in the *Sensor ID* column of the list view. How incoming values are processed is specified by the target type. Details are explained in the table below.

Target type	The target type defines how tracking values are used within VRED. The values are: <ul style="list-style-type: none"> • <i>Custom Node</i>: Tracking values are directly transferred to the Node specified in the <i>Target Node</i> field. • <i>Head Projection</i>: Tracking values are transferred to the camera and used for power-wall and cave projections. • <i>HMD</i>: Tracking values are transferred to the camera and used for head mounted displays. • <i>Cart</i>: Tracking values are transferred to the camera used for cart scenarios.
Target node	Only accessible for target type <i>Custom Node</i> . Specifies the node the tracking values will be transferred to.
Translation offset	Possibility to add an extra translation in local body coordinates.
Rotation offset	Possibility to add an extra rotation.

Cart transform mode	Only accessible for target type <i>Cart</i> . Specifies the source where the cart will get its values from. The values are: <ul style="list-style-type: none"> • <i>Custom</i>: Translation and rotation values are gathered from the <i>Cart Translation</i> and <i>Cart Rotation</i> fields. • <i>Node</i>: Transform values are gathered from a node in the scene graph. The name of that node is specified in the <i>Cart Node</i> field. • <i>Viewpoint</i>: Transform values are computed by a viewpoint. The name of the viewpoint is specified in the <i>Viewpoint</i> field.
Cart node	Only accessible for cart transform mode <i>Node</i> . Specifies the name of the node in the scene graph. If the node does not exist, this value will be ignored.
Viewpoint	Only accessible for cart transform mode <i>Viewpoint</i> . Specifies the name of the viewpoint which is used to compute the transformation. If the viewpoint does not exist, this value will be ignored.
Ignore device offsets	If checked, the offset values from the device will be ignored for final position and orientation computations.

Fly-stick settings

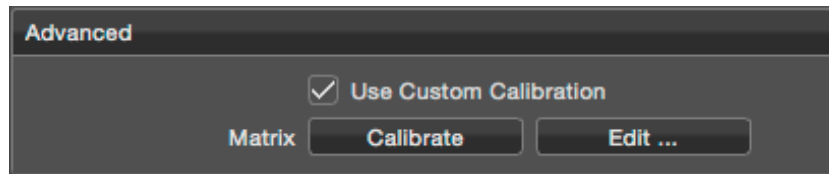


A fly-stick is a special kind of body with a predefined button and analogs. It acts as a navigation device and is therefore used to change the users position and orientation within the virtual scene. There are two different modes: *Scene Move* and *Point Fly*. Additionally it is possible to define buttons and analogs and connect those via script to specific functions in VRED (see python script example).

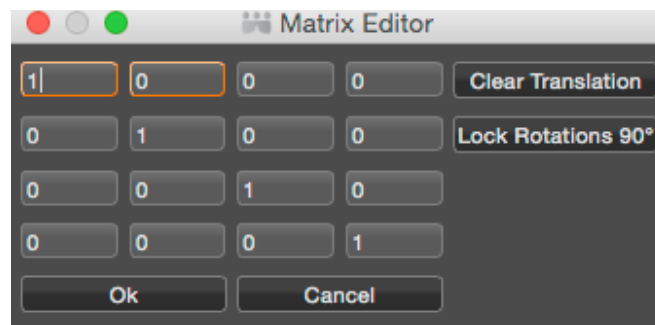
Mode	Sets the interaction mode for the fly-stick. The values are: <ul style="list-style-type: none"> • <i>None</i>: No interaction active. • <i>Scene Move</i>: Move the scene with the fly-stick. It is necessary to press the related button to activate this mode. • <i>Point Fly</i>: Moves the user to pointed direction when the analog stick is pushed.
Speed	Possibility to adjust the speed for movements in point fly mode.
Translation offset	Possibility to add an extra translation in local body coordinates.
Rotation offset	Possibility to add an extra rotation.

Ignore device offsets	If checked, the offset values from the device will be ignored for final position and orientation computations.
Show position	Shows an orange cube to highlight the fly-stick position.

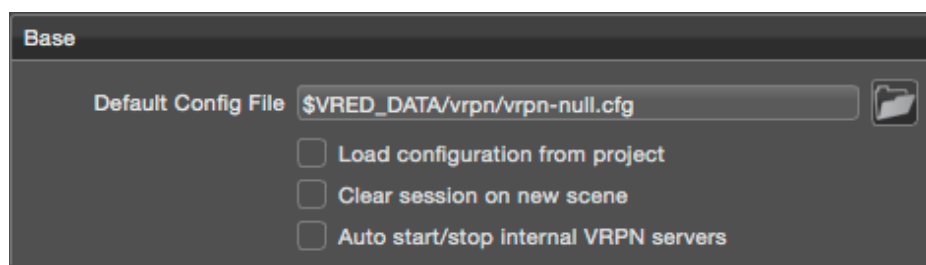
Advanced settings



There is the possibility to specify a custom modification matrix for incoming tracking values of devices, bodies and fly-sticks. By enabling this option, all other settings will be ignored. The matrix can be manually modified by clicking the *Edit* button or for bodies and fly-sticks automatically by executing *Calibrate*. The latter will use the current position of the sensor as a reference.



Preferences



Default config file	Specifies the default VRPN configuration file. It is used for server configuration.
Load configuration from project	If enabled the current configuration may be overwritten with the one from a loaded project.
Clear session on new scene	If enabled the current configuration will be cleared for a new scene.
Auto start/stop internal VRPN servers	If enabled, internal VRPN servers will be automatically started/stopped when enabling/disabling tracking.

Python API

The VRPN tracking module in VRED supports the access over the Python API. Following list shows the currently supported functions.

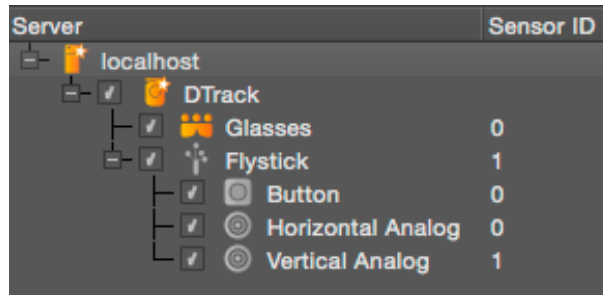
loadTrackingConfig(string)	Loads a tracking configuration from an xml file.
startTracking()	Enables tracking.
stopTracking()	Disables tracking.
bool isTracking()	Returns true if tracking is enabled, otherwise false is returned.
setFreezeTracking(bool)	Enables/disables a tracking freeze.
bool getFreezeTracking()	Returns true if tracking is freezed, otherwise false is returned.
setTrackingUnitScale(float)	Sets the tracking unit scale. Default is 1000. VRPN normally delivers the data in metres, VRED expects millimetres.
vrTrackerButton(string)	Creates a tracker button object defined by the passed name string. Functions that can be called on it: <i>setSensorId(int)</i> and <i>bool isPressed()</i> . Signals: <i>pressed</i> and <i>released</i> .
vrTrackerAnalog(string)	Creates a tracker analog object defined by the passed name string. Functions that can be called on it: <i>setSensorId(int)</i> and <i>double getValue()</i> .

Examples

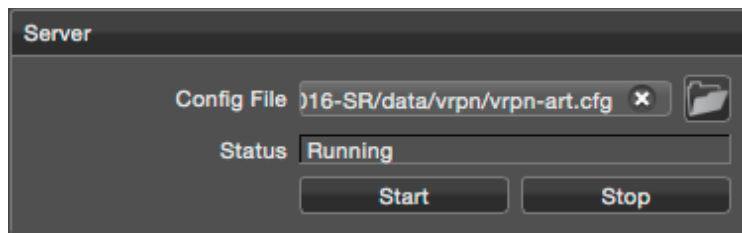
How to set up an A.R.T tracking configuration

Assuming the A.R.T. room calibration was done with the setting *Power wall*. This implies that the tracking coordinate system is right-handed y-up orientated. The origin of it is on the ground and there is a front wall projection of size 3x2 metres which is orthogonal to the ground. Furthermore no VRPN server is currently running.

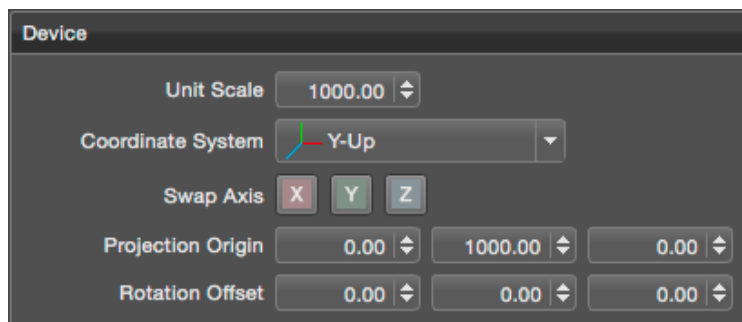
The following steps are an example for how to create a configuration for a projection with head-tracking (glasses body) and fly-stick navigation.



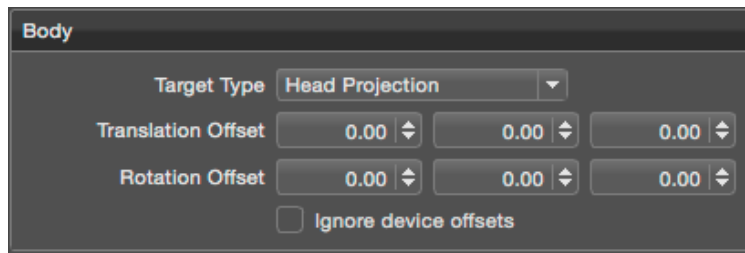
1. Create a new VRPN server. The name should be *localhost* since a new server will be started.
2. Load the tracking configuration file *vrpn-art.cfg* from the example data folder.
3. Start the server.



4. Add a new device with name *DTrack* to the server. This corresponds to the name in the configuration file.
5. Change the coordinate system to *Y-Up* because the room was calibrated with setting *Power wall*. There is no need to swap any axis.
6. Set the y-value of the projection origin to 1000. This is half the diameter of the front wall. We assume the projection is set up symmetrical to its size.



7. Add a new body to the device (it may be necessary to change the sensor identifier).
8. Change its target type to *Head Projection*.
9. Additionally, if the tracker sensor for the glasses was calibrated with *zero in marker*, you can add a translation offset to move the reference point in the middle of the glasses.



10. Add a new fly-stick to the device (it may be necessary to change the sensor identifier). Automatically a button and two analogs are created. They will be needed for the different navigation modes.
11. Change the navigation mode to *Scene Move* or *Point Fly*. First one uses the predefined button of the fly-stick to activate the movement. The latter one uses the two analogs for a fly-to navigation.



12. Optionally you can check the configuration with a connection test. It can be found in the diagnose menu.
13. Last step: Activate tracking by toggling the enable/disable button in the bottom toolbar.



Python script for button interaction

The following python script shows a simple example of how to connect fly-stick buttons to specific VRED functionality. It is based on an existing tracking configuration with two buttons. Those were created with the names *Button1* and *Button2*. If the buttons would have equal names, it is additionally necessary to specify the sensor identifier.

```
def myFunction():
    print "Test!"

b1 = vrTrackerButton("Button1")
b1.connectSignal("pressed", myFunction)

b2 = vrTrackerButton("Button2")
b2.connectSignal("released", showInitialView)
```

Trouble shooting

Nothing happens when tracking is enabled

This may have several causes. Please check the following things:

- Is a VRPN server running on the specified host? If not, you can start one within the server settings frame.
- Is the tracking system sending its data to the correct host (where the VRPN server is running on)?
- Is the device name correct? The device name has to match with the one in the VRPN configuration.
- Is the sensor identifier correct and is the sensor in tracking range?

VRPN server is not starting

If you have problems starting a VRPN server, it is possible that the VREDClusterService process is not running.

Problems with the new user interface

If you experience problem with the new user interface, it is currently possible to switch back to the old one by adding the environment variable `VRED_USE_OLD_TRACKING` with a value of 1. This option is only temporary and will be removed in one of the next releases.