

2021

VOTANICXR

VotanicXR Configurator

USER MANUAL

VOTANIC LIMITED

Revision Sheet

Table of Contents

<i>VotanicXR Configurator</i>	4
<i>VotanicXR Configurations</i>	4
<i>VotanicXR Configurator Overview</i>	5
Configuration File Operations	5
Create New Configuration File	7
Open an Existing Configuration File	8
Save the Configuration File	9
Revert Changes in the Configuration File	12
Switch VotanicXR Configurator Appearance	13
XR System Configuration	14
The Environment Types	14
System Configuration Overview	16
3D Space View	17
Viewports Diagram	18
Configure System	19
Configure Device	21
Device List View	22
General Device Properties Setting	23
Add a New VRPN Device	24
Configure VRPN Tracker	25
Configure VRPN Button	27
Configure VRPN Axis	28
Test Auto Scanned Devices	29
Use Virtual Trackers	30
Configure XR Objects	31
Hierarchy Overview	32
XR Objects in System Config Templates	33
Add a New XR Object Node to Hierarchy	35
Configure Node	36
Configure Entity	37
Configure Screen	39
Configure Camera	40
Configure Display	42
Viewports Diagram	43
Add and Configure a Display	43
Add and Configure a Viewport	44
Configure Info Wall	45

<i>Application Settings</i>	47
Application Settings Overview	47
Add a Scene Profile	48
Configure XR Attributes	50
Configure Performance Properties	51
Configure Frame & User	53
Add a New User	53
Configure Frame Transform	54
Configure User Transform	57
Configure Controllers	61
Add a New Controller	62
Configure a Controller	62
Configure Controller Tools	63
Configure Commands	65
Add a New Command	66
Configure Command Properties	67
Command Trigger – Virtual Controller Input Event Mapping	70
<i>Application Options Setting</i>	73
Application Options Setting Overview	73
Configure In-Game Menu	74
Add / Remove In-Game Menu Item	75
Configure In-Game Menu Global Setting	75
Configure Menu Panel	80
Configure Menu Button	81
Configure Landing Menu	83
Configure Landing Menu Global Setting	84
Configure Waypoints	91

VotanicXR Configurator

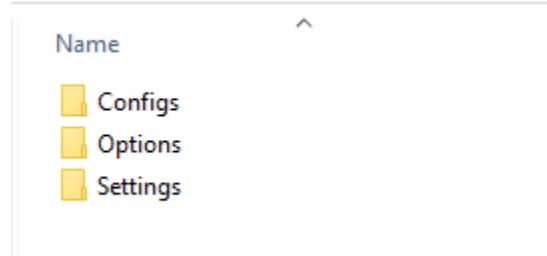
The VotanicXR Configurator offers a user-friendly graphical user interface to configure various settings of the VotanicXR application. Here we provide an overview to the basic terminology in configuring VotanicXR application and customizing the application's environment for running the same VotanicXR application across different XR systems.

VotanicXR Configurations

VotanicXR is designed around the concept of being cross-platform, and the configuration files are VotanicXR's way to achieve the cross-platform compatibility. By customizing the configurations, one can create different configurations and settings for the same VotanicXR application so that it can run smoothly on different XR systems.

In general, VotanicXR separates the configurations into two main categories: XR System Configuration file, with file extension “.vxrc”, which configures mainly the physical setup of the XR system; and the Application Configurations, with file extension “.vxrs” and “.vxro”, which configure the application setting and in-game menu, respectively.

By default, the configuration files are stored in the “Assets/Votanic/VotanicXR/vGear/System” directory in the Unity project folder, and in the “Build/VotanicXR” directory in the build folder.



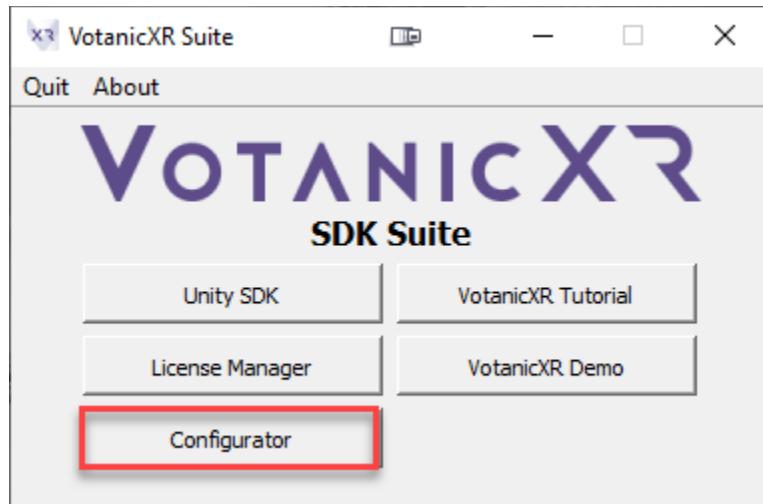
The XR System Configuration files (e.g. “ConfigHMD.vxrc” file for a HMD XR System) are stored in the “Configs” folder. These config files define the physical configurations of a XR system, such as the tracker / sensors available in the system and physical input devices connected to the system.

The Application Setting files (i.e. “Setting.vxrs” file) is stored in the “Settings” folder. The file is mainly responsible for various application-specific settings such as the input commands and quality settings.

The Application Options Setting file (i.e. “Option.vxro” file) is stored in the “Options” folder. This file manages advanced application-specific settings such as the in-game menu and the landing page settings.

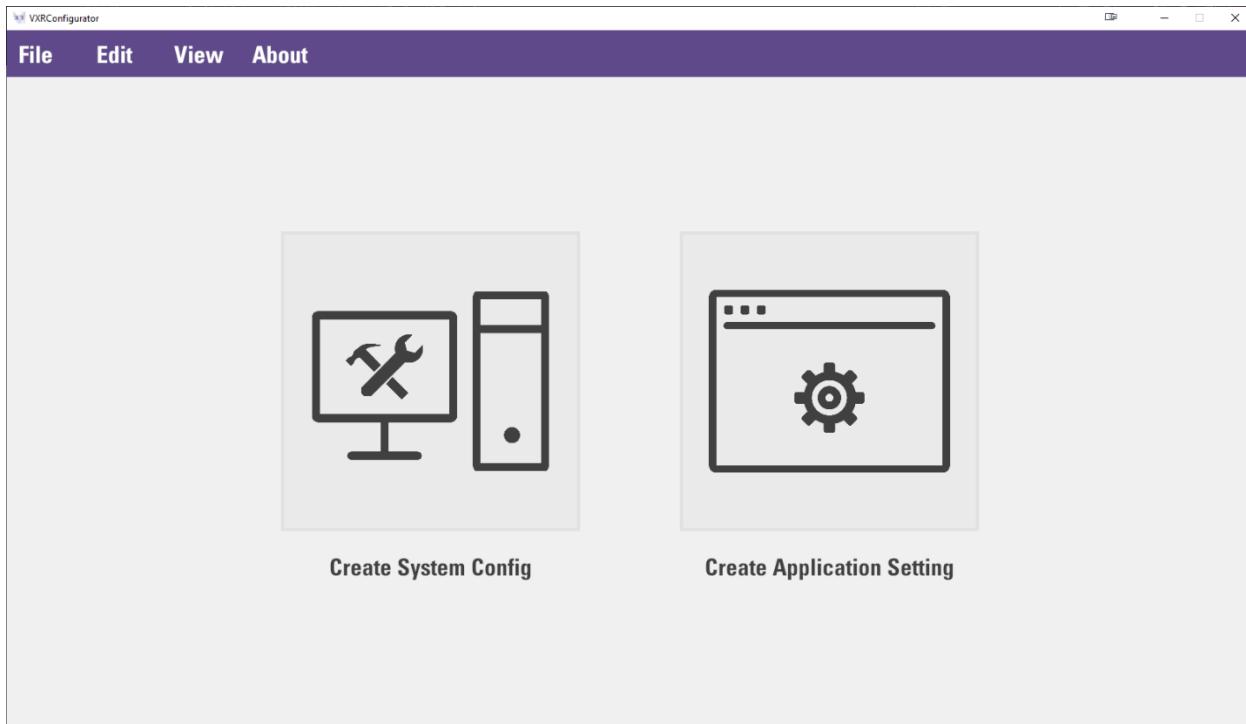
VotanicXR Configurator Overview

The VotanicXR Configurator can be opened by selecting the “Configurator” button from the VotanicXR Suite or simply double-click and open a VotanicXR configuration file with a file extension “.vxrc”, “.vxrs”, or “.vxro”.

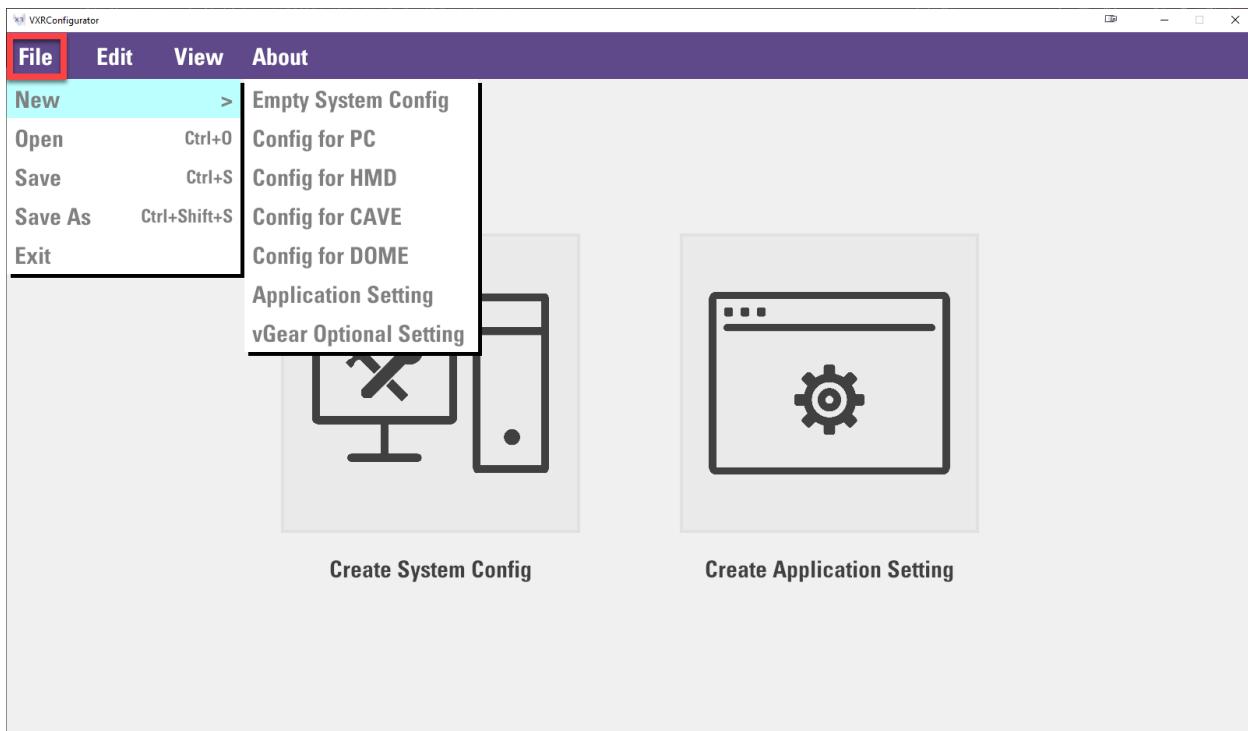


Configuration File Operations

If you open the VotanicXR Configurator from the VotanicXR Suite, you will see the landing page of the configurator. You can create a new configuration file by clicking the “Create System Config” button or the “Create Application Setting” button.

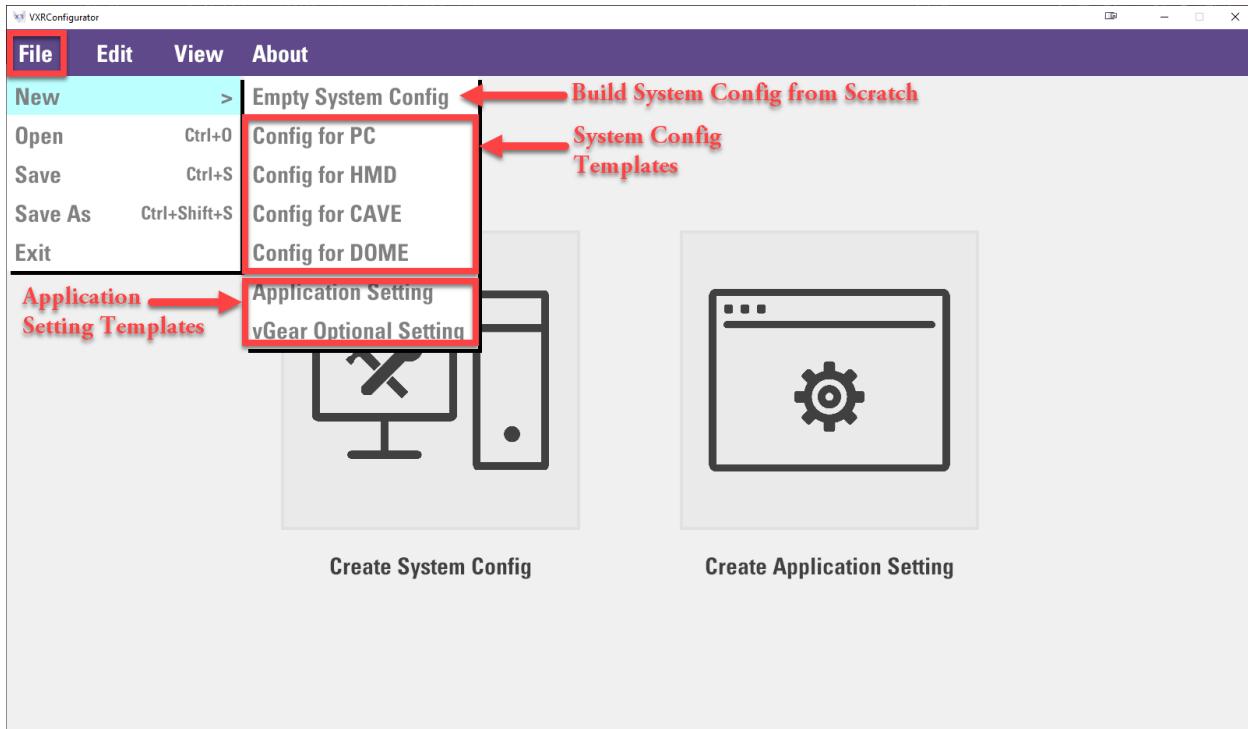


The “File” menu can be used to perform various configuration file related operations.



Create New Configuration File

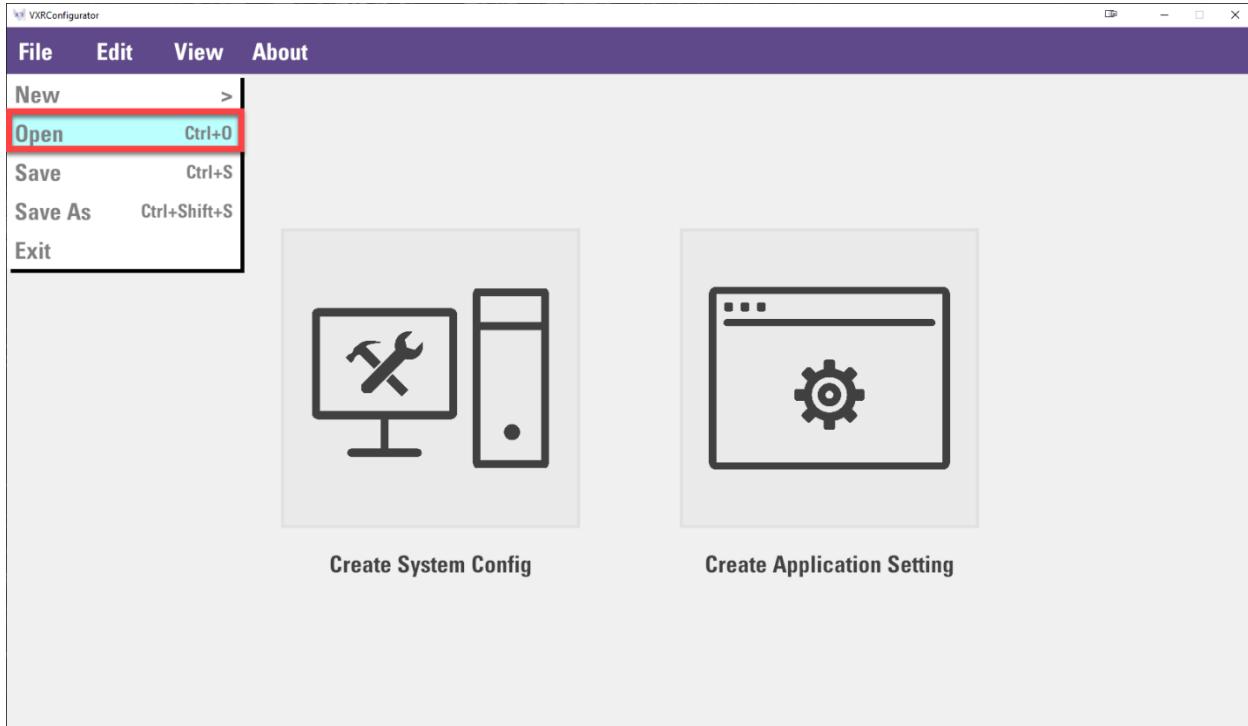
You can use either the “Create Environment Config” / “Create Application Setting” button, or the “File -> New” options menu to create a new configuration file.



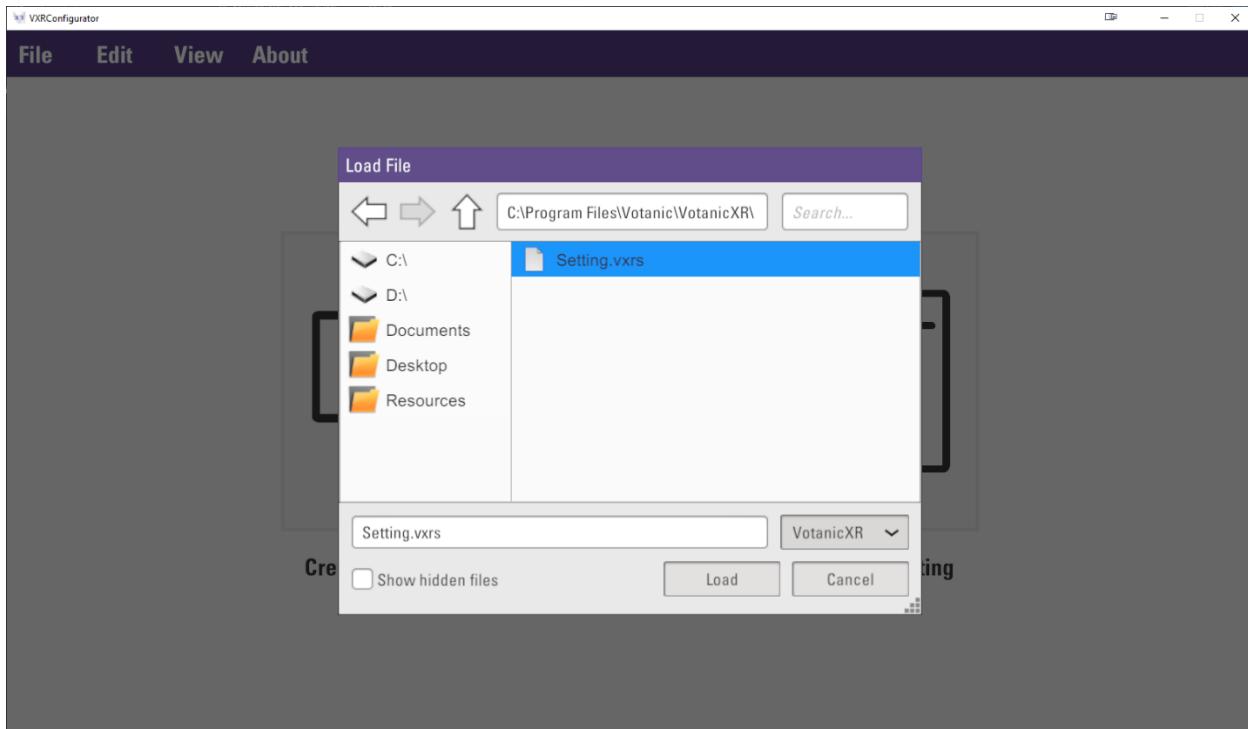
There are several template files available to choose as a starting point to create your own configuration file. Choose a template file that best suits your usage to start building the configuration file.

Open an Existing Configuration File

Select “File -> Open” to select an existing configuration file in your file system to open.

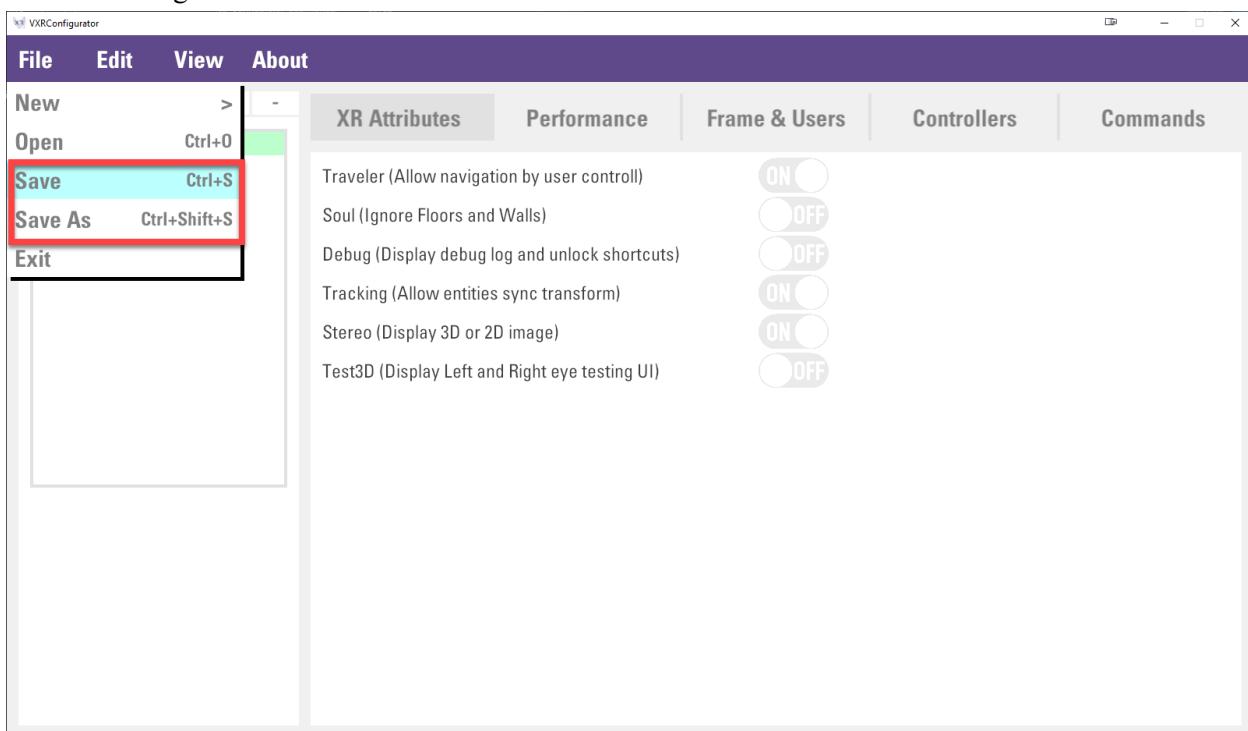


A file browser will be displayed. In the file browser, navigate to the directory with the VotanicXR configuration file, then select the desirable file and click “Load”. The corresponding configurator page will be loaded and presented to you.

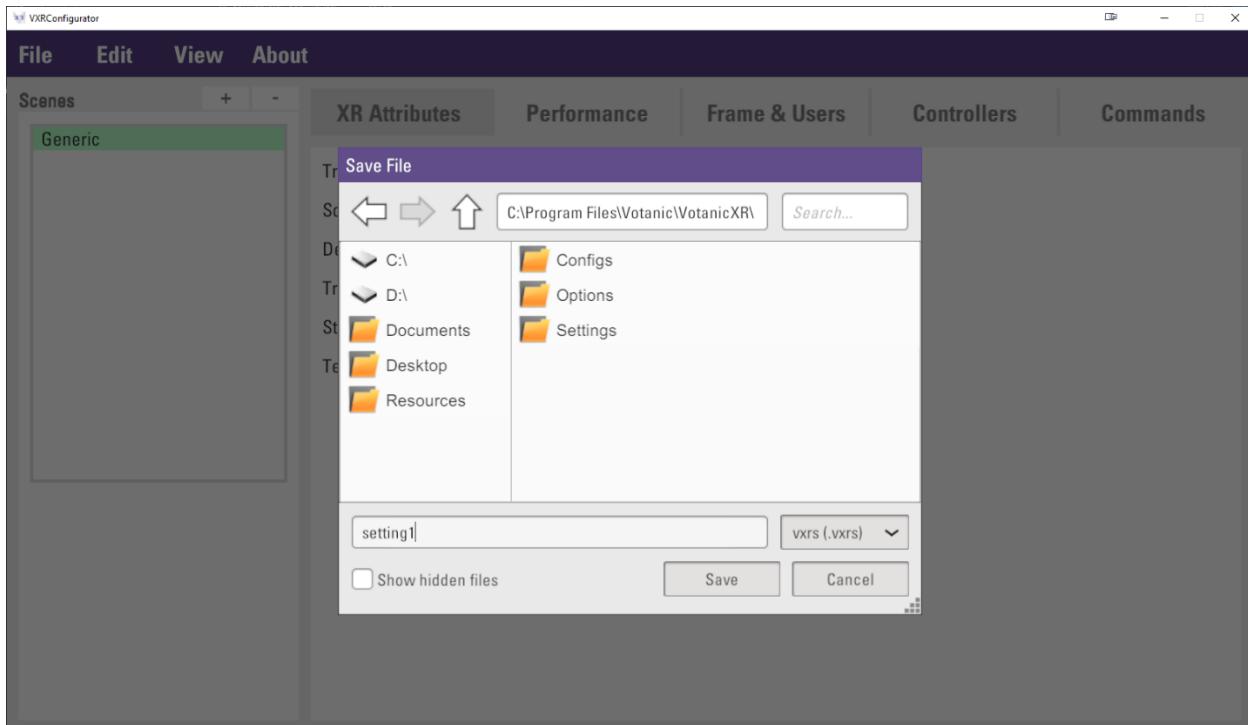


Save the Configuration File

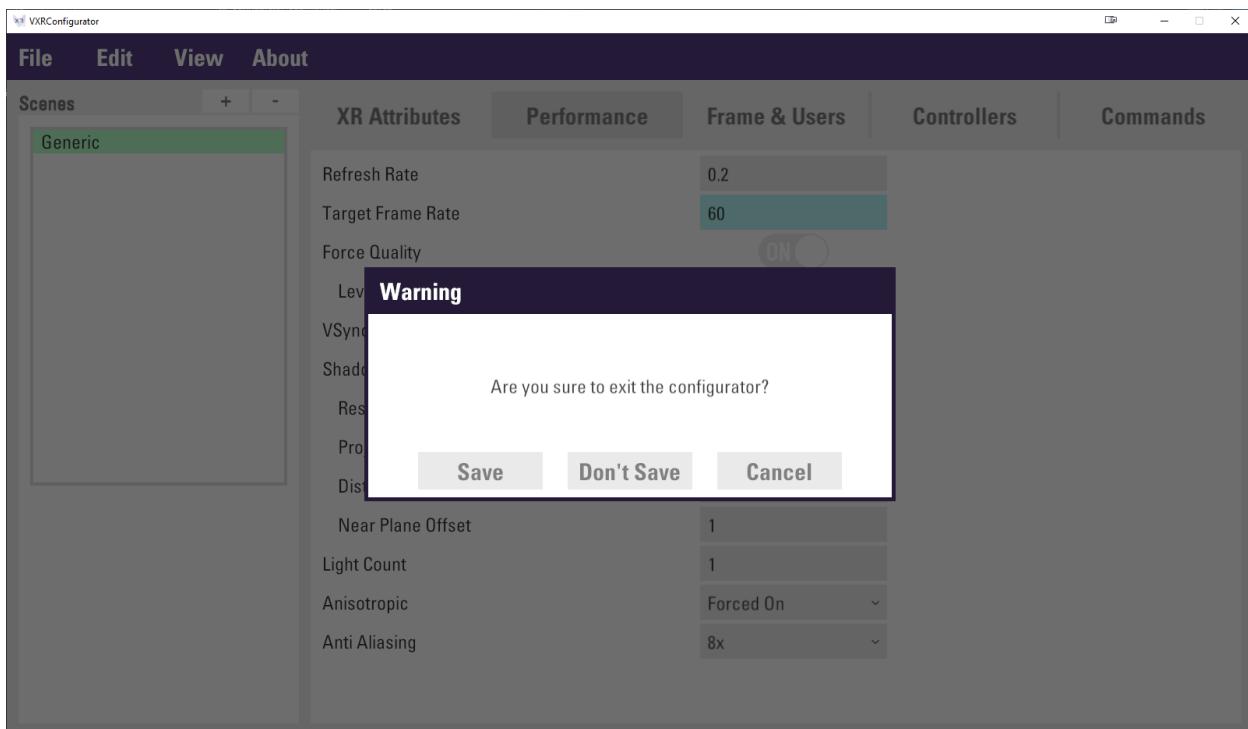
Use “File -> Save” to save your changes to the existing file or “File -> Save As” to save the configuration to a new configuration file.



The file browser will appear if it is the first time you select “File -> Save” after creating a new configuration file, or if you have chosen “File -> Save As”. Choose an appropriate directory to save the file.



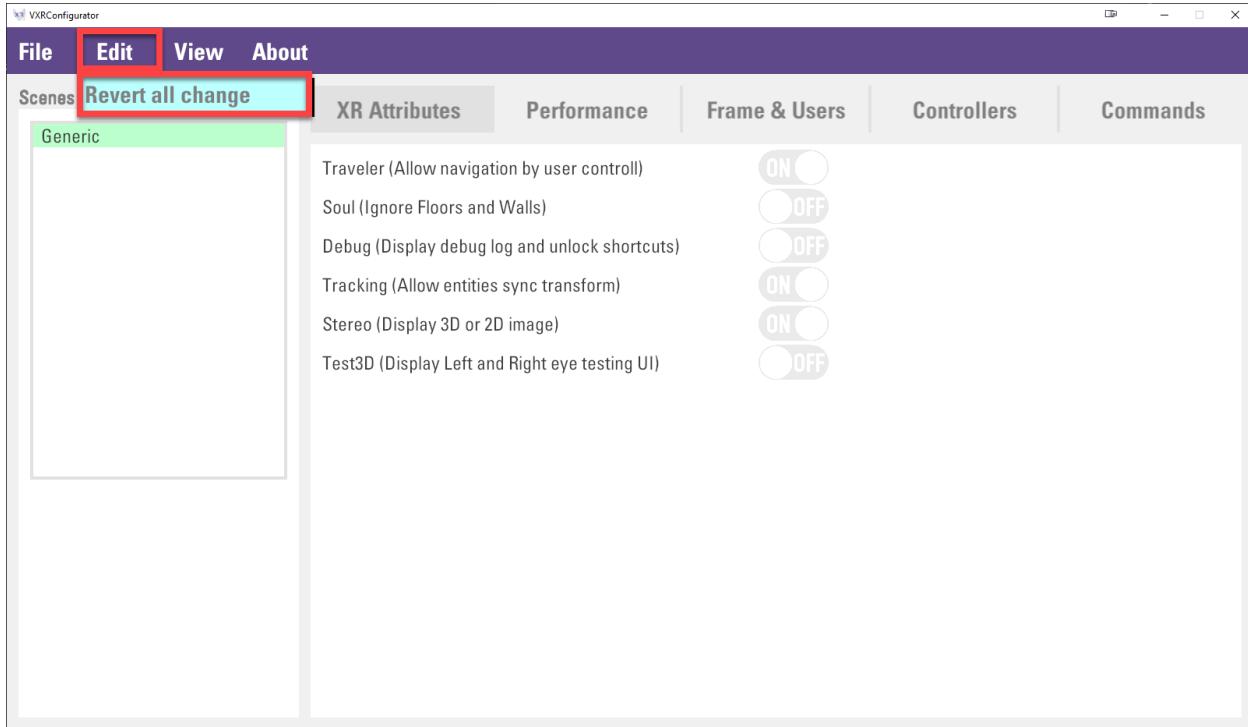
If you try to close the VotanicXR Configurator with unsaved changes, a warning will be shown prompting you to save your changes.



Note: If you have SteamVR installed on your computer, SteamVR will recognize the VotanicXR Configurator as a SteamVR application. If you have the configurator open while trying to launch another SteamVR application (regardless of whether it is created using VotanicXR), SteamVR will force close the VotanicXR Configurator and you will lose all changes made in the configurator. So, be sure to save and close the VotanicXR Configurator first before you launch another SteamVR application.

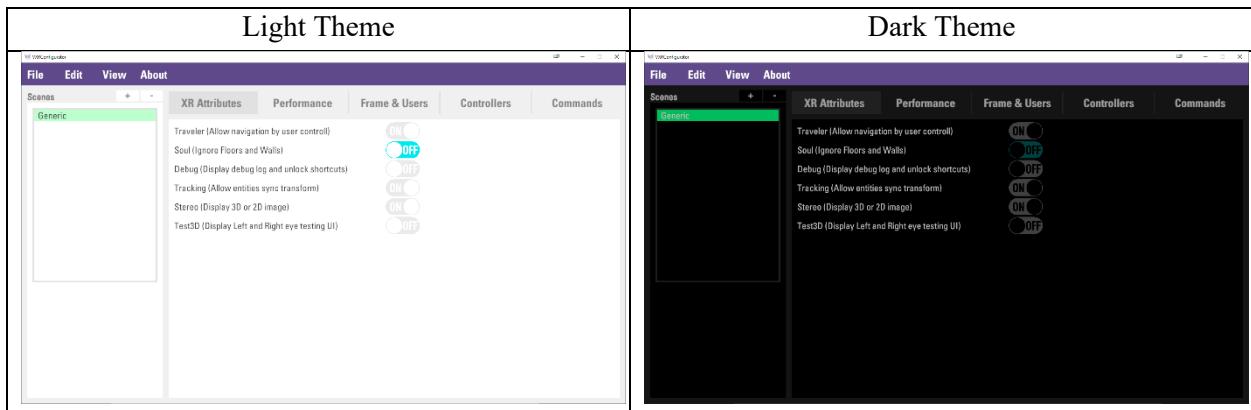
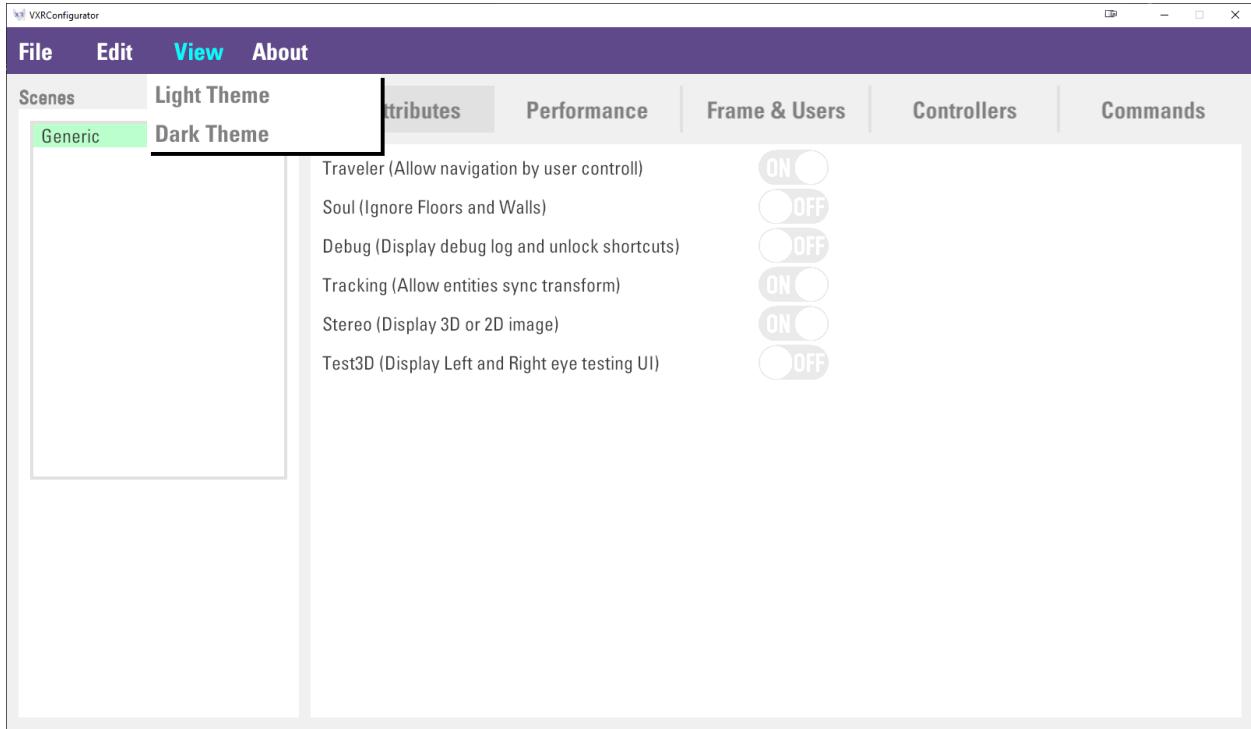
Revert Changes in the Configuration File

You can use the “Edit -> Revert all change” option to **undo all changes** you made in the configuration file back to the state of the last save.



Switch VotanicXR Configurator Appearance

You can use the “View” menu to change color theme of the VotanicXR Configurator.



XR System Configuration

The XR system configuration files, with file extension “.vxrc”, defines the configuration of the XR system that is running the VotanicXR application. In principle, a single system configuration file is sufficient for a XR system with a fixed physical configuration, all VotanicXR applications can then use this system configuration file to launch the application in its correct XR specification.

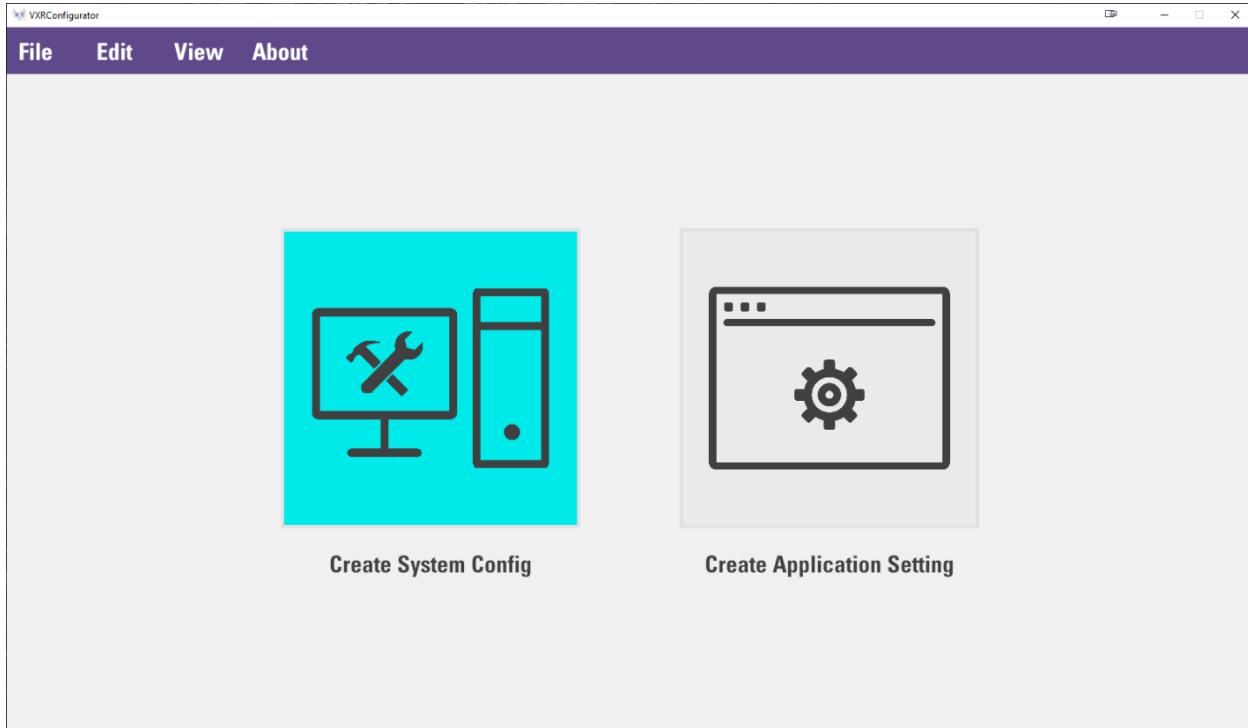
The Environment Types

VotanicXR has built-in features that adapts to different XR system environments. It is necessary to define the environment type of the configuration so that the VotanicXR feature can function as expected during runtime. The supported environment types are as follows:

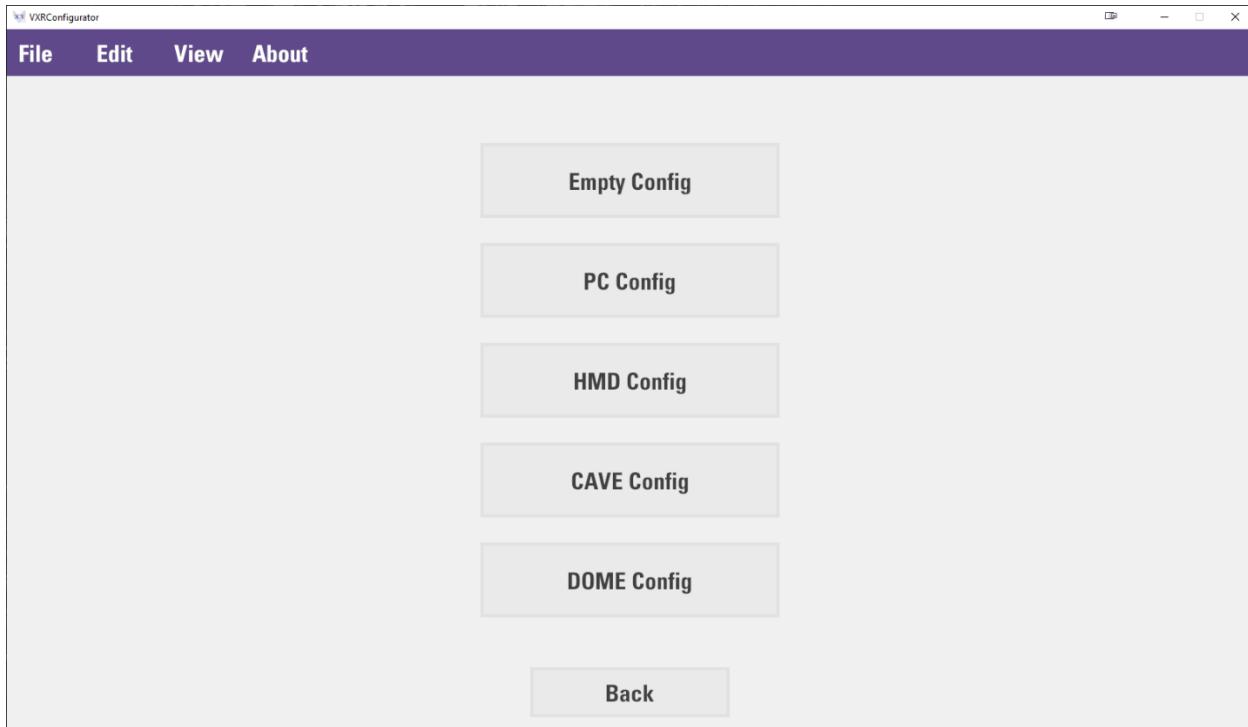
Environment Type	Description
PC	The system is a PC with a single fixed resolution display, having inputs received from Keyboard and Mouse. Most settings are pre-defined and not modifiable.
HMD	The system is a head-mounted display (HMD) VR system, tracking and input are received from the SteamVR driver. Most settings are pre-defined and not modifiable.
CAVE	The system is a multiple display system. The configuration of the CAVE system is highly flexible with every aspect of the configuration modifiable.
DOME	The system is an immersive dome system, having inputs received from Keyboard and Mouse. Most settings are pre-defined and not modifiable.

Note: *VotanicXR player license is required for changing the environment setting in runtime. Apart from the CAVE environment type, using the default configuration template is usually sufficient for the other environment types under typical usage.*

To create a new system configuration file, select “Create System Config” on the VotanicXR Configurator’s landing page.

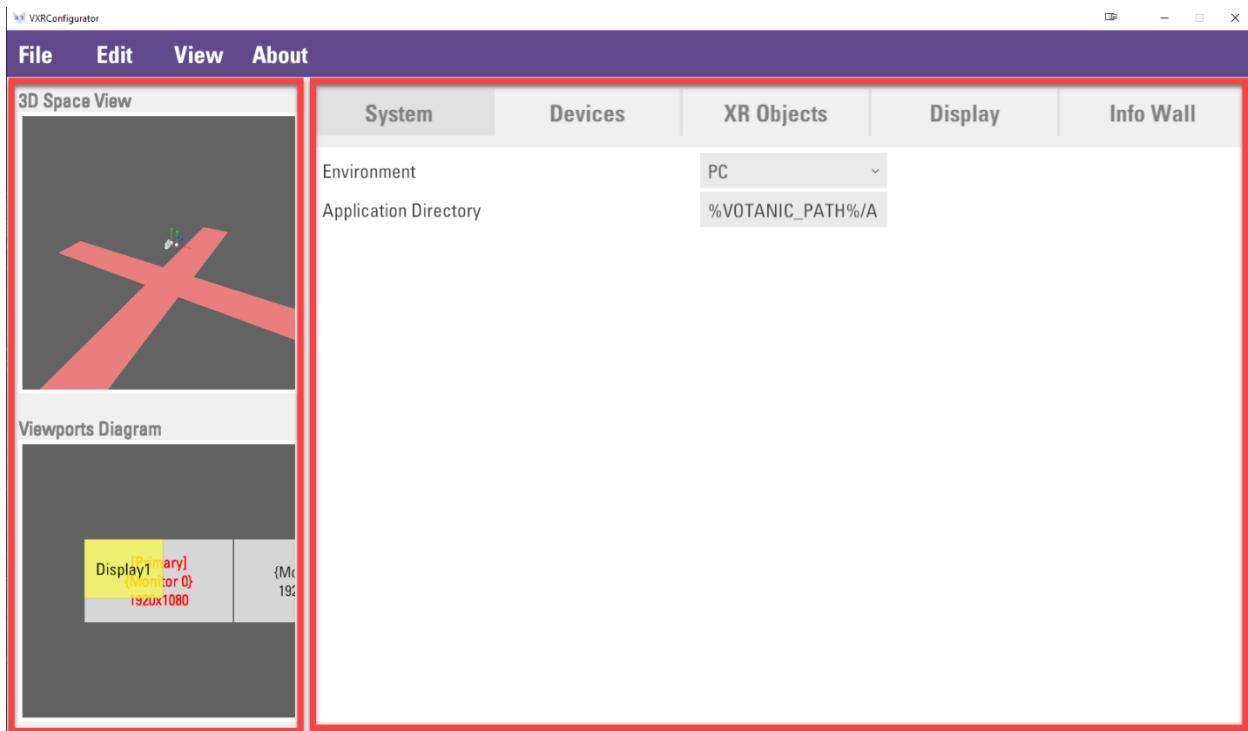


Select an appropriate system configuration template to use as a starting point, or start from scratch by choosing the “Empty Config” option, to create the configuration file.



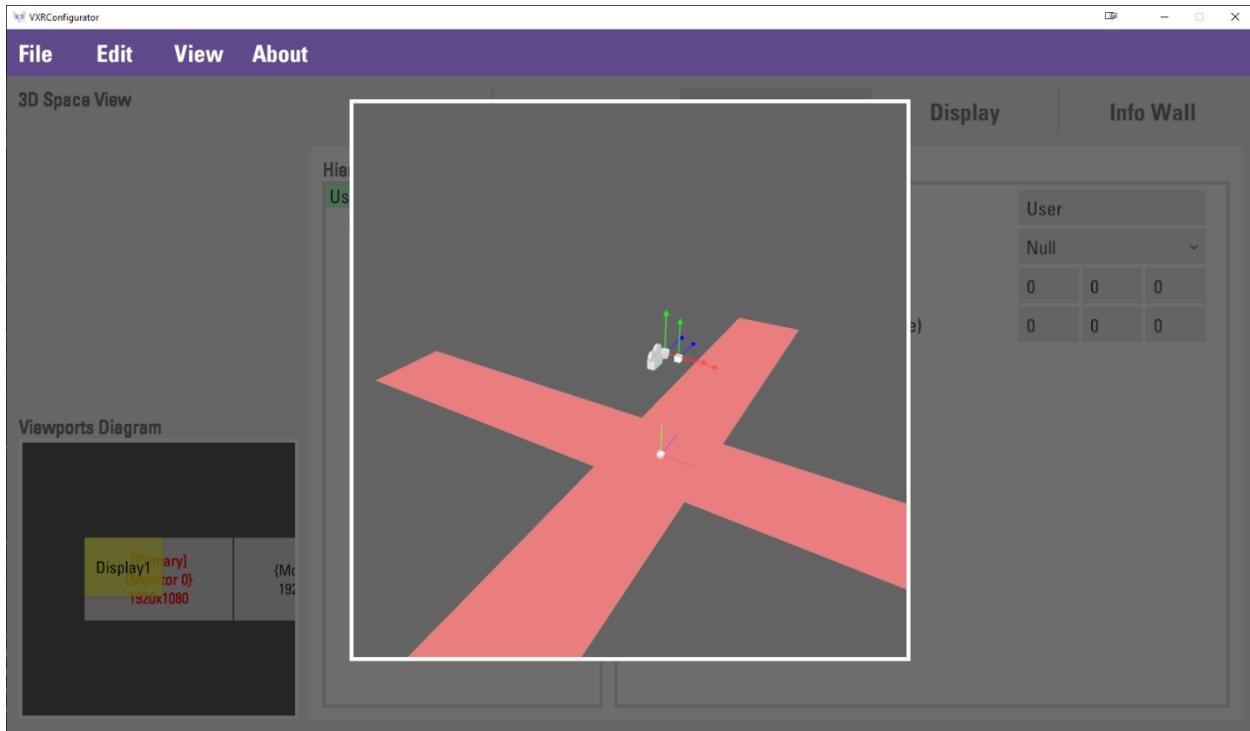
System Configuration Overview

The system configuration view in the VotanicXR Configurator is mainly divided into two segments. The large area on the right-hand side stores different properties of the system configuration, separated into different groups for ease of accessing different properties. The left-hand side of the system configuration view shows both a visualization of the 3D space of the XR system as well as the display rendered by the VotanicXR application; both the 3D Space View and the Viewport Diagram are updated in real time as you update the configuration.



3D Space View

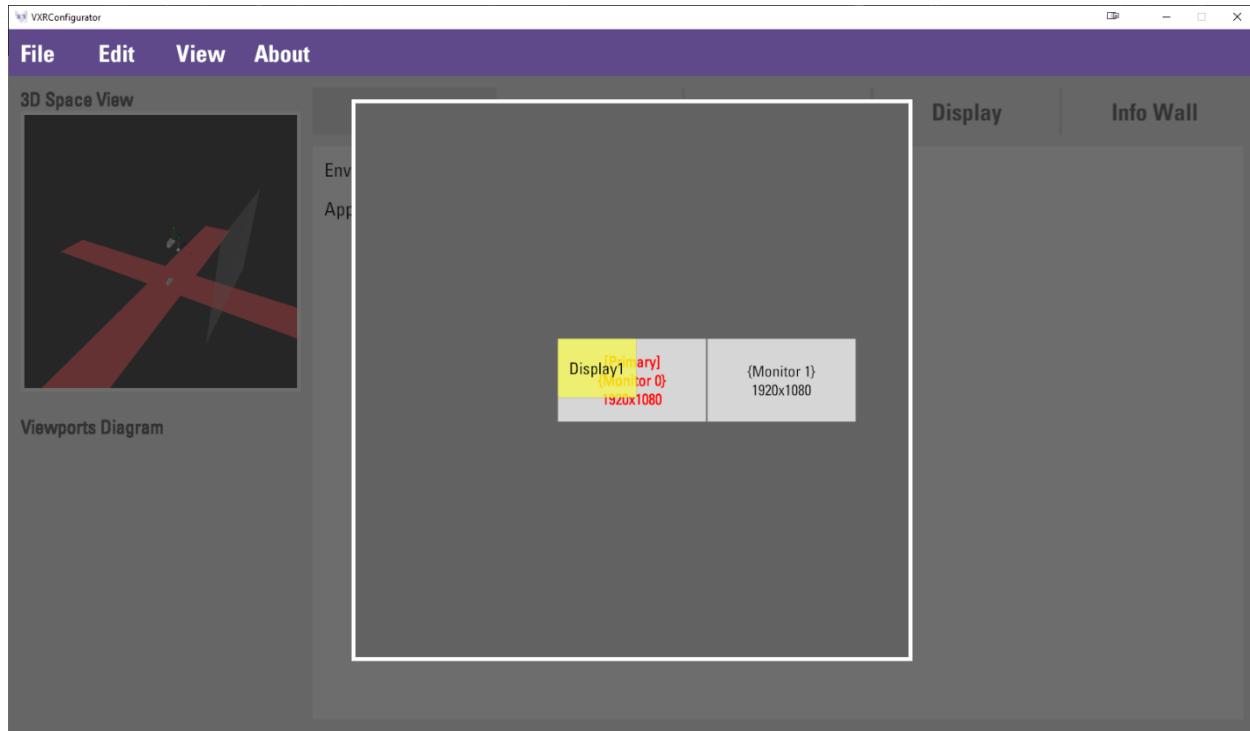
The 3D Space View visualizes how different components of the physical environment you defined in the configuration will look like in 3D. You can click on the 3D space view to enlarge it. Use the mouse wheel to zoom in/out the 3D space, right-click and drag to rotate, hold the middle button and drag to pan, and use the left click to close the enlarged view.



More details of the 3D Space View will be discussed in the “Configure XR Objects” part of this System Configuration guide.

Viewports Diagram

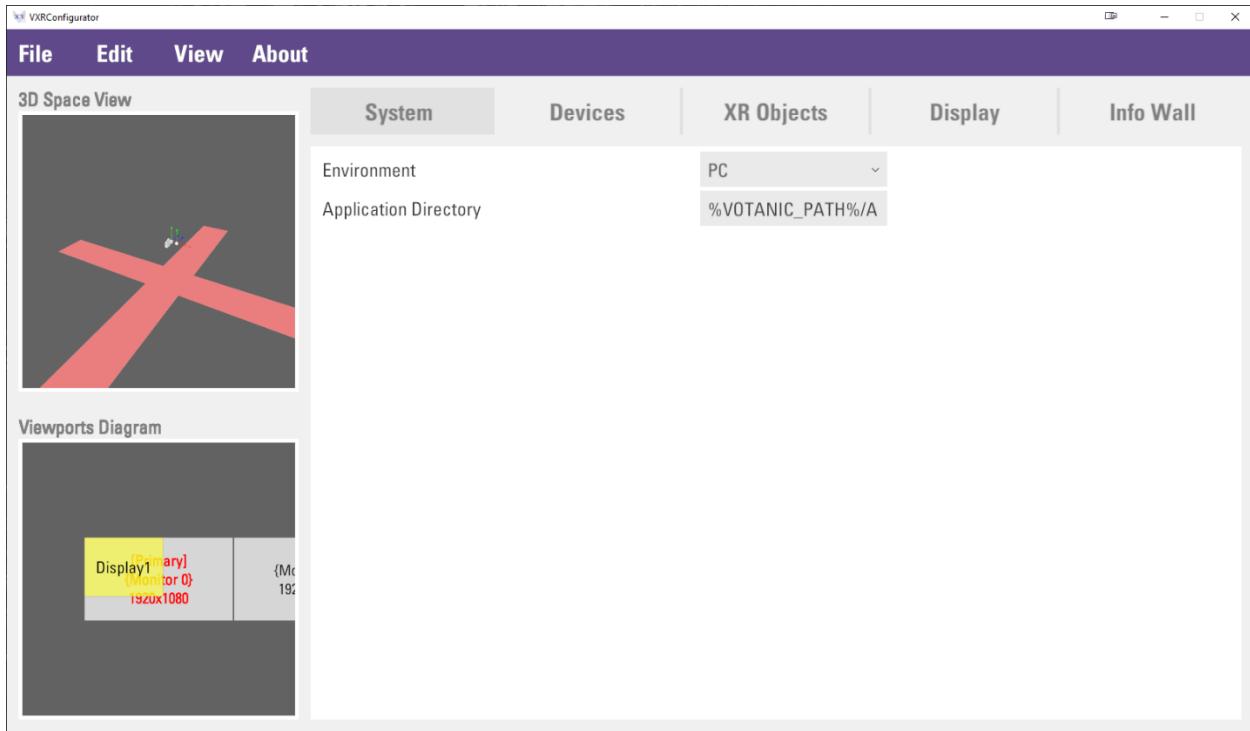
The viewports diagram visualizes how the Display and Viewports you defined in the configuration will show up on the physical monitor. You can click on the viewports diagram to enlarge it.



More details of the Viewports Diagram will be discussed in the “Configure Display” part of this Environment Configuration guide.

Configure System

The “System” properties mainly relate to the general system setting of the XR system.



The table below describes the specifics of the “System” properties:

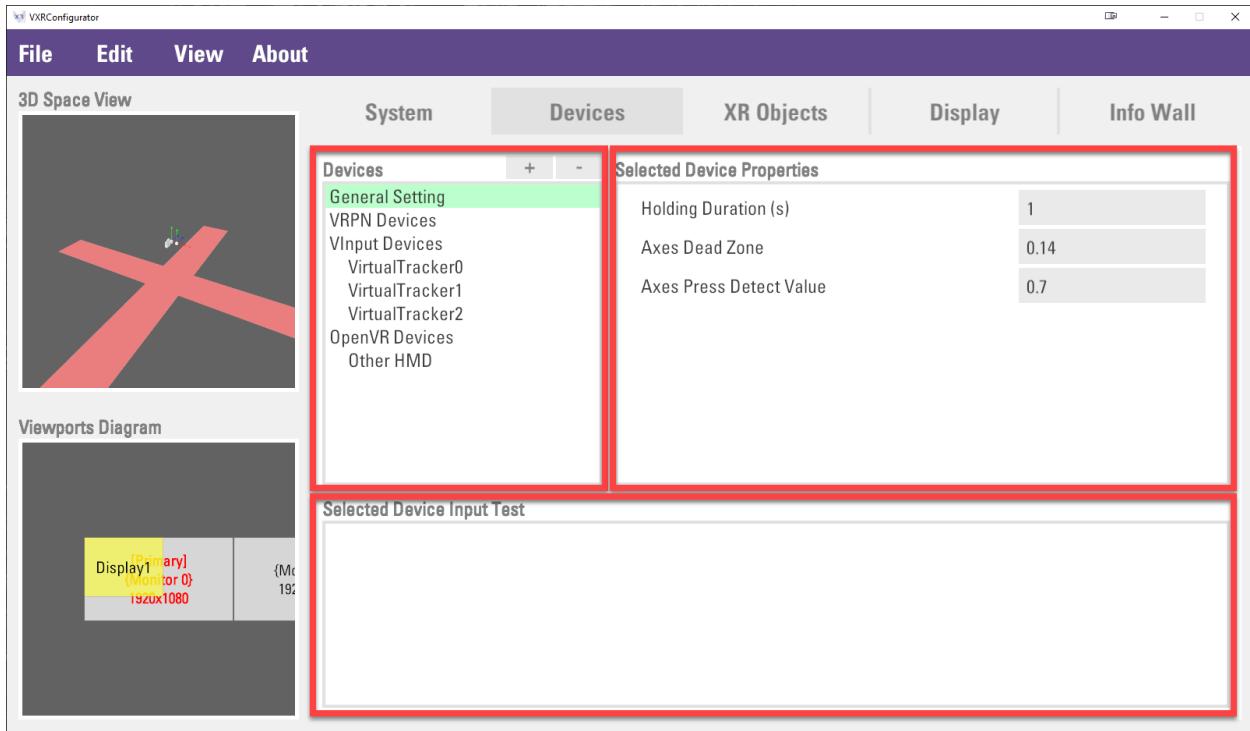
System Property	Description
Environment	<p>Defines the XR system environment for the System Configuration file. VotanicXR application will adapt to behave differently based on the different “Environment” setting.</p> <p>The available “Environment” options are:</p> <ul style="list-style-type: none">• PC• HMD• CAVE• DOME
Application Directory	<p>Defines the directory path to “Launcher.json” file which indicated other VotanicXR applications location. If “Launcher.json” file is existed, the list of application will be displayed in the “Apps” in-game menu. User can switch between VotanicXR applications in the application.</p> <p>The example of “Launcher.json” is shown in below code block</p>

Launcher.json sample:

```
{  
  "Category": [  
    {  
      "Name": "Media Player",  
      "Application": [  
        {  
          "Name": "Media",  
          "Path": "path/to/the/application.bat"  
        },  
        {  
          "Name": "Media360",  
          "Path": "path/to/the/application.bat"  
        }  
      ]  
    },  
    {  
      "Name": "Project",  
      "Application": [  
        {  
          "Name": "Project ABC",  
          "Path": "path/to/the/application.bat"  
        }  
      ]  
    }  
  ]  
}
```

Configure Device

The “Devices” properties of the XR system can be configured in the Devices tab.



In general, the Devices tab is divided into 3 sections:

- The **devices list view** at the top-left portion displaying a list of devices that is currently connected to the XR system, or VRPN devices which has been manually added to the system.
- The **device properties view** at the top-right portion displaying properties of the selected device that are available for configuration.
- The **device input test view** at the bottom portion displaying the various states of the selected device. It is useful when testing the status of the device.

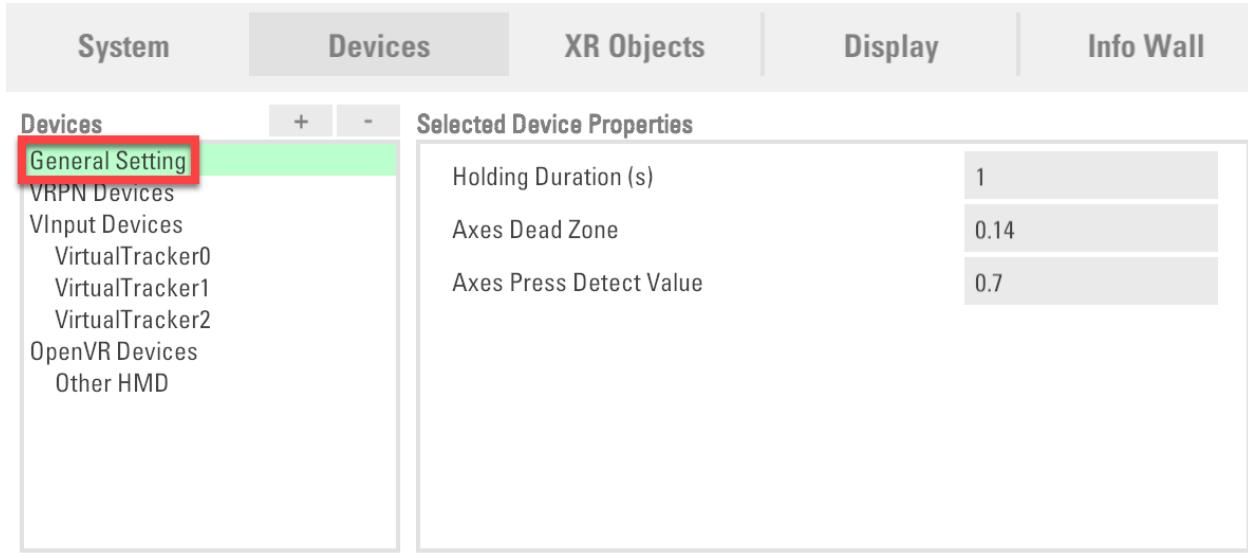
Device List View

The device list view shows a list of devices in the XR system. In general, there are three types of devices which will show in the device list:

Device Type	Description	Show in List	Remarks
VRPN Devices	<p>Networked devices streaming from a VRPN server.</p> <p>Supported VRPN device type:</p> <ul style="list-style-type: none">• VRPN Tracker• VRPN Button• VRPN Analog	Manually added to the device list	Requires a VRPN server set up and running to obtain data from these VRPN devices.
VInput Devices	Devices using the XInput or Direct Input driver that are currently turned on and connected to the system.	Automatically shown in the device list if the device is turned on and connected to the system	
OpenVR Devices	VR devices currently turned on and recognized by SteamVR on the system. The VR device can be a SteamVR device, an Oculus device, or a windows mixed reality device, as long as SteamVR recognizes it.	Automatically shown in the device list if the device is turned on and connected to the system	Requires SteamVR to be installed and running in the system to obtain data from these OpenVR Devices

General Device Properties Setting

General device properties can be modified by selecting the “General Setting” item at the top of the device list view:

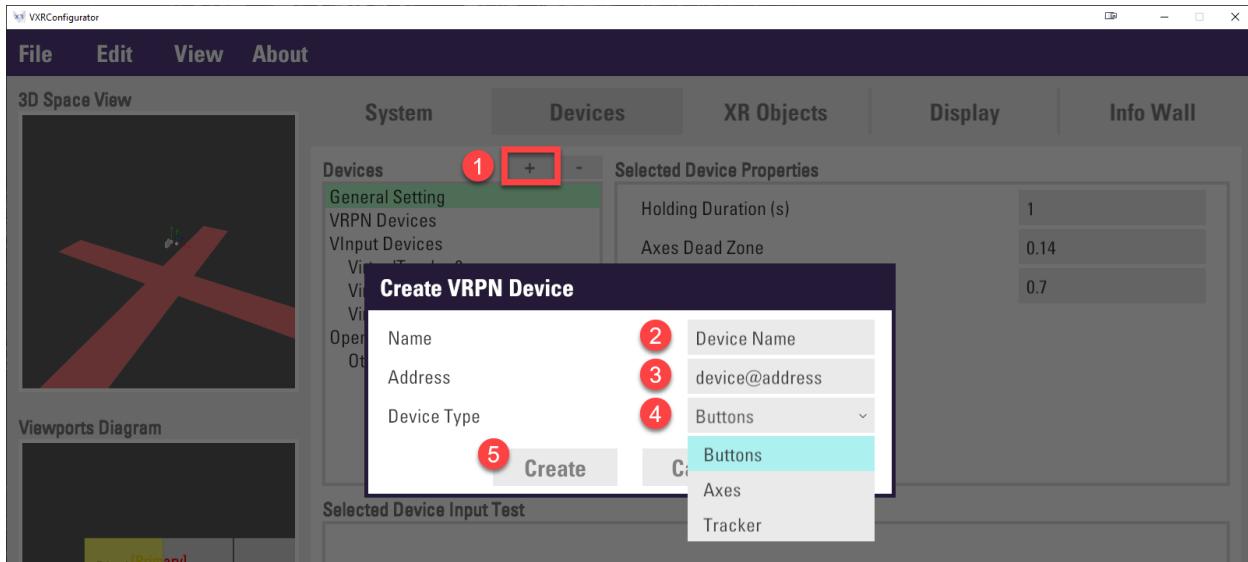


The general setting defines the global device properties for all devices that are automatically added to the device list, i.e. the **VInput Devices & OpenVR Devices**, the settings include:

Device Property	Description
Holding Duration(s)	Float value in seconds. Holding the button for longer than the specified value will be treated as “Button Hold”.
Axes Dead Zone	Float value between 0 and 1. Axis value smaller than the specified value will be treated as 0.
Axes Press Detect Value	Absolute float value between 0 and 1. Axis value greater than the specified value will invoke a “Axes Press” button event.

Add a New VRPN Device

VRPN devices are networked devices whose states are broadcasted via a VRPN server. You can add a VRPN device to the Environment Configuration, which enables VotanicXR to use the VRPN device in VotanicXR applications.



To add a new VRPN device, click the “+” button in the device list view, then in the Create VRPN Device panel, input the “Name”, “Address”, and select the “Device Type” for the new VRPN device. Details of the Create VRPN Device panel properties are explained below:

VRPN Device Property	Description
Name	<p>Self-defined name that can uniquely identify the VRPN device.</p> <p>Note: <i>If you add more than one device using the same name, only the first device in the device list will be accessible by VotanicXR.</i></p>
Address	<p>Network address of the VRPN device streamed from the VRPN server.</p> <p>Typically, the address of the VRPN device uses the following convention: “devicename@NetworkAddress:PortNumber”</p> <p>Note: <i>The port number can be omitted if the VRPN server is using the default port (i.e. 3883)</i></p>
Device Type	<p>Define the type of the VRPN device. The supported VRPN device types are:</p> <ul style="list-style-type: none">• Buttons: return button states of an array of buttons on the VRPN button device• Axes: return axis values of an array of axes on the VRPN analog device• Tracker: return position and rotation of the VRPN tracker device

Configure VRPN Tracker

For every VRPN tracker added to the VRPN device list, it is necessary to configure the VRPN tracker so that the tracker behaves correctly in VotanicXR application.

The screenshot shows the VotanicXR configuration interface with the 'Devices' tab selected. On the left, a tree view shows 'General Setting', 'VRPN Devices', 'VInput Devices' (with 'VirtualTracker0', 'VirtualTracker1', 'VirtualTracker2'), 'OpenVR Devices', and 'Other HMD'. A 'Tracker1' node under 'VRPN Devices' is selected and highlighted in green. To the right, the 'Selected Device Properties' panel displays the following settings for 'Tracker1': Name: Tracker1, Address: HeadTracker@192.168, Channel: 0, Forward: +Z, Up: +Y, Right: -X. Below this, the 'Selected Device Input Test' panel shows the current state of the 'Tracker Position' and 'Tracker Rotation'.

Selected Device Input Test
Tracker Position -0.840 1.514 -1.555 Tracker Rotation 28.108 216.792 356.202

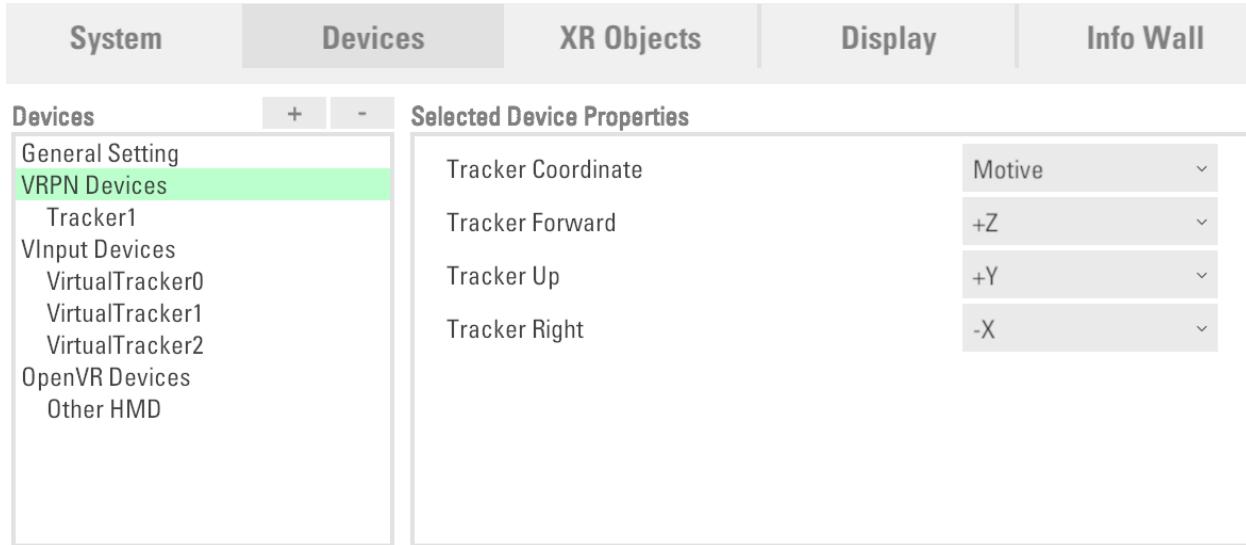
After adding the VRPN tracker, select the tracker that you have just created and configure the tracker channel and coordinate system in the “Selected Device Properties” view. Details of the VRPN tracker properties are explained below:

VRPN Device Property	Description
Channel	The channel index in which the VRPN tracker is streamed from the network address of the VRPN tracker.
Forward / Up / Right	The Forward / Up / Right direction defined in the VRPN tracker's coordinate system. For example, if the VRPN tracker's X-axis is pointing to the left, you need to select “-X” in the dropdown list for the “Right” direction.

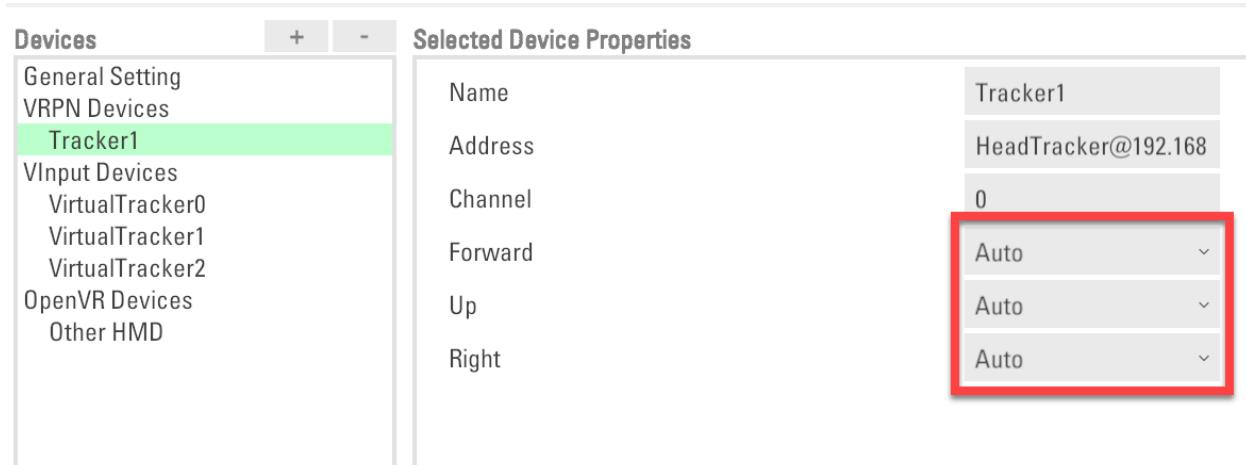
If the VRPN tracker is set up correctly, the tracker states (i.e. tracker position and tracker rotation) will be fetched from the VRPN server and displayed in the “Selected Device Input Test” view. The tracker position and tracker rotation displayed in the VotanicXR configuration and used in VotanicXR applications are using “+Z / +Y / +X” as the “Forward / Up / Right” direction.

Define VRPN Tracker Coordinate System Globally

If you add a number of VRPN trackers all using the same coordinate system, it is possible to define the coordinate system of all trackers globally, in the VRPN Devices setting.



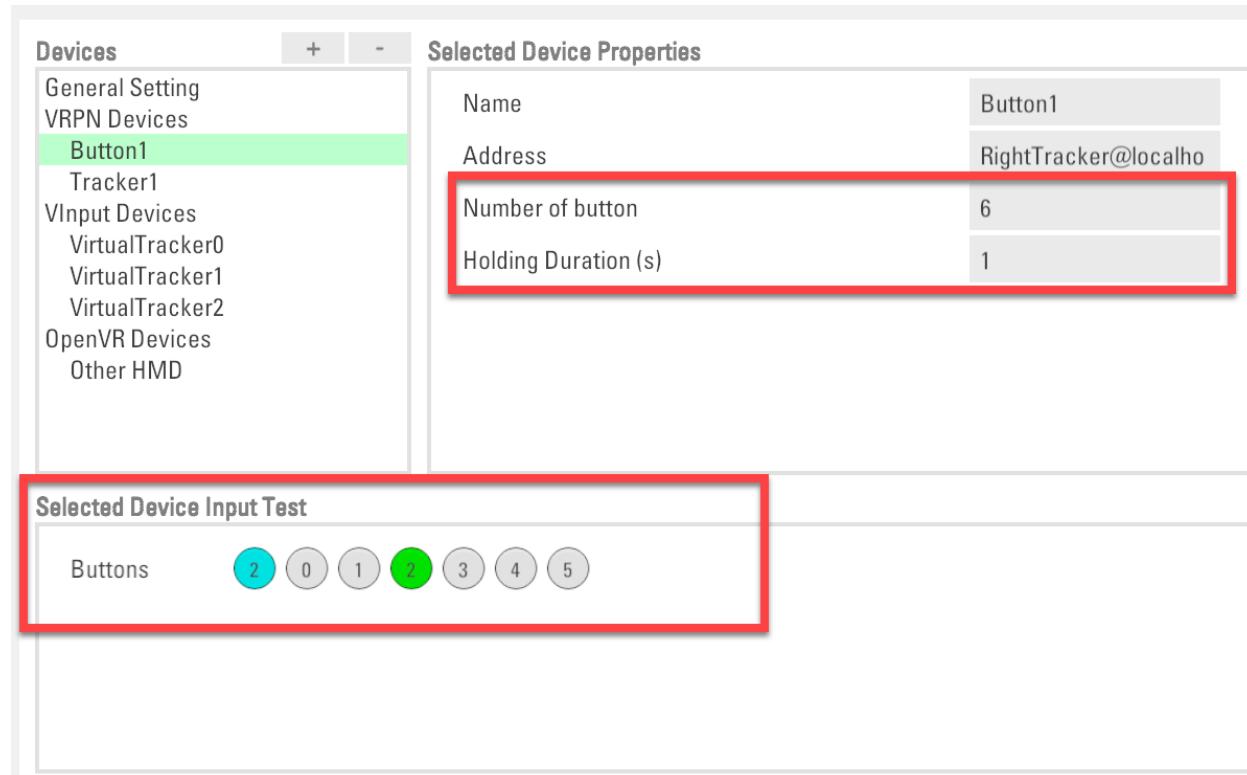
Select “VRPN Devices” from the device list view, then you will be able to configure the coordinate system for all VRPN trackers in the selected device properties view. The “Tracker Coordinate” property provides a list of coordinate system presets for commonly used software.



Set the “Forward / Up / Right” direction of a VRPN Tracker to “Auto” to use the global coordinate system setting defined in the VRPN Devices properties.

Configure VRPN Button

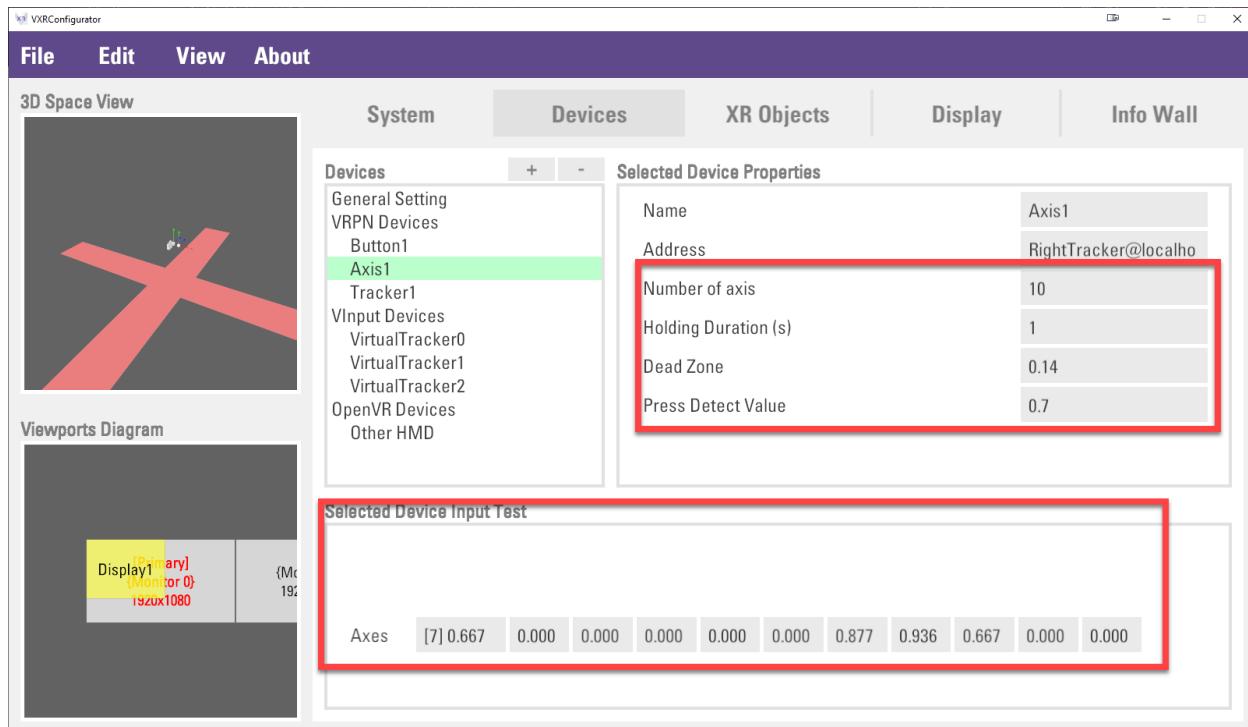
After adding the VRPN button device, select the newly added VRPN button device and define the number of buttons usable on the device. Optionally, you can also customize the holding duration for the VRPN button device.



The “Selected Device Input Test” view can be used to test the states of the buttons on the selected VRPN button device. In the example above, the input test area would display buttons with indexes based on the number of buttons defined for the device. The button index will change colour when you press the button of the corresponding button index.

Configure VRPN Axis

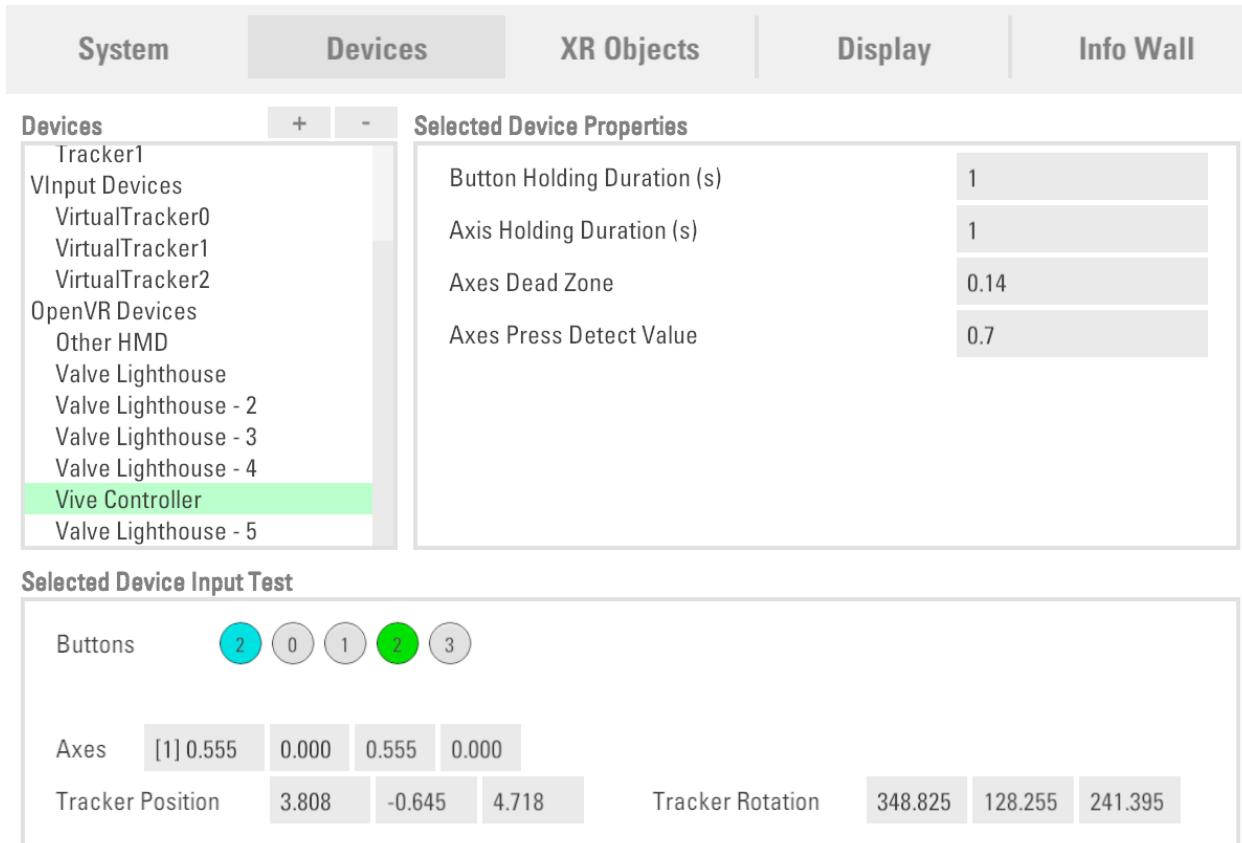
After adding the VRPN axis device, select the newly added VRPN axis device and define the number of axis usable on the device. Optionally, you can also customize the holding duration (for pressing and holding the axis to invoke the axis hold event), axis dead zone value, and axis press detect value (for pressing the axis to invoke the axis press event) the for the VRPN axis device.



The “Selected Device Input Test” view can be used to test the values of the axes on the selected VRPN axis device. In the example above, the input test area would display axes with indexes based on the number of axis defined for the device. When you move the axes on the device, the axis value for the corresponding axis in the device would change.

Test Auto Scanned Devices

The VotanicXR Configurator automatically scans for available devices when the configurator starts. All automatically scanned devices are added to the device list, and you can select the device to check the device's all available states.



Selected Device Properties	
Button Holding Duration (s)	1
Axis Holding Duration (s)	1
Axes Dead Zone	0.14
Axes Press Detect Value	0.7

Selected Device Input Test					
Buttons	2	0	1	2	3
Axes	[1] 0.555	0.000	0.555	0.000	
Tracker Position	3.808	-0.645	4.718	Tracker Rotation	348.825 128.255 241.395

In the above example, the vive controller has 4 buttons, 3 axes, as well as a tracker status on it, they are all displayed and updated in real-time.

Note: SteamVR requires the HMD to connect to the system before the button and axes events are streamed out. Be sure an HMD is available before testing the SteamVR-powered controllers.

Use Virtual Trackers

The virtual trackers are simple tracker devices which imitate the rotation and height translation of a tracker.

System Devices XR Objects Display Info Wall

Devices

+ -

Selected Device Properties

Button Holding Duration (s) 3

Axis Holding Duration (s) 3

Axes Dead Zone 0.14

Axes Press Detect Value 0.7

Selected Device Input Test

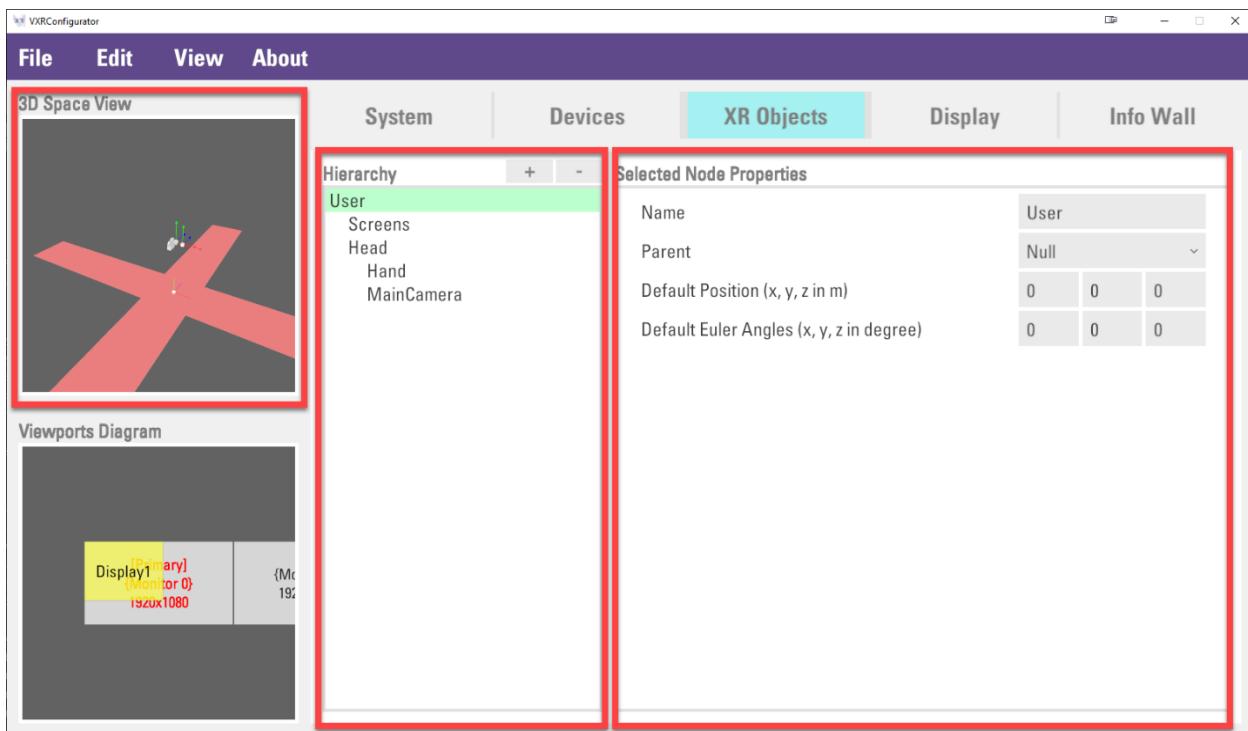
Tracker Position 0.000 0.800 0.000 Tracker Rotation 40.200 355.300 0.000

There are 3 virtual trackers provided natively by VotanicXR, the typical use is to attach them to the user's head use them to rotate the user's head when debugging the application on PC when no tracking is available. The controls of the virtual trackers are summarized below:

Virtual Tracker	Change Tracker Rotation (X-Axis and Y-Axis only)	Change Tracker Position (Y-Axis only)
VirtualTracker0	Rotate X-Axis: move mouse forward / backward Rotate Y-Axis: move mouse left / right	Scroll the mouse wheel forward and backward
VirtualTracker1	Rotate X-Axis: hold mouse right button + move mouse forward / backward Rotate Y-Axis: hold mouse right button + move mouse left / right	Hold mouse right button + scroll the mouse wheel forward and backward
VirtualTracker2	Rotate X-Axis: hold mouse center button + move mouse forward / backward Rotate Y-Axis: hold mouse center button + move mouse left / right	Hold mouse center button + scroll the mouse wheel forward and backward

Configure XR Objects

The XR Objects tab defines the main components of the XR system, as well as how different components work together to produce the desirable behaviour in the VotanicXR application. During runtime, the VotanicXR application would adapt the behaviour of the application based on the Hierarchy defined in XR Objects.



In general, the XR Objects tab can be divided into two views:

- **Hierarchy View:** defines the components of the XR system and their relationships.
- **Node Properties View:** displays and configures the properties of a selected node in the Hierarchy.

Based on the hierarchy defined in the Hierarchy view and the corresponding node properties, the 3D Space view would be rendered in real-time to visualize the main components and their relationships defined in the Hierarchy.

Hierarchy Overview

The “Hierarchy” defines the components of the XR system (XR objects) and their relationships. During runtime, the XR objects defined in the Hierarchy are synchronized to the VotanicXR application and then used accordingly by the application to generate different XR experiences. The basic components of XR objects are described below:

XR Object Type	Description
Node	<p>Node is a generic XR Object in the Hierarchy. It can be understood as an empty GameObject in Unity’s Hierarchy that is used to group its child objects.</p> <p>All Node objects has a “Name”, a “Parent”, and a transform (in terms of “Position” and “Rotation”). “Entity”, “Camera”, and “Screen” are also Node objects with added domain-specific properties.</p>
Entity	<p>Entity is a special type of Node with Tracking and Input properties. User can attach a tracker and an input controller to an Entity such that the Entity moves along with the tracker, and making the Entity accessible to various in-game features of VotanicXR.</p>
Camera	<p>Camera is a special type of Node with added properties related to camera setting. The Camera properties provide a number of configurations which govern the behaviour of the Camera in the VotanicXR application.</p>
Screen	<p>Screen is a special type of Node with added properties specific to a Screen. “Screen” here refers to rectangular-shaped physical surface in the XR system that is used to display rendered image. A Screen works in conjunction with a Camera, which renders its corresponding camera viewports on the screen.</p>

XR Objects in System Config Templates

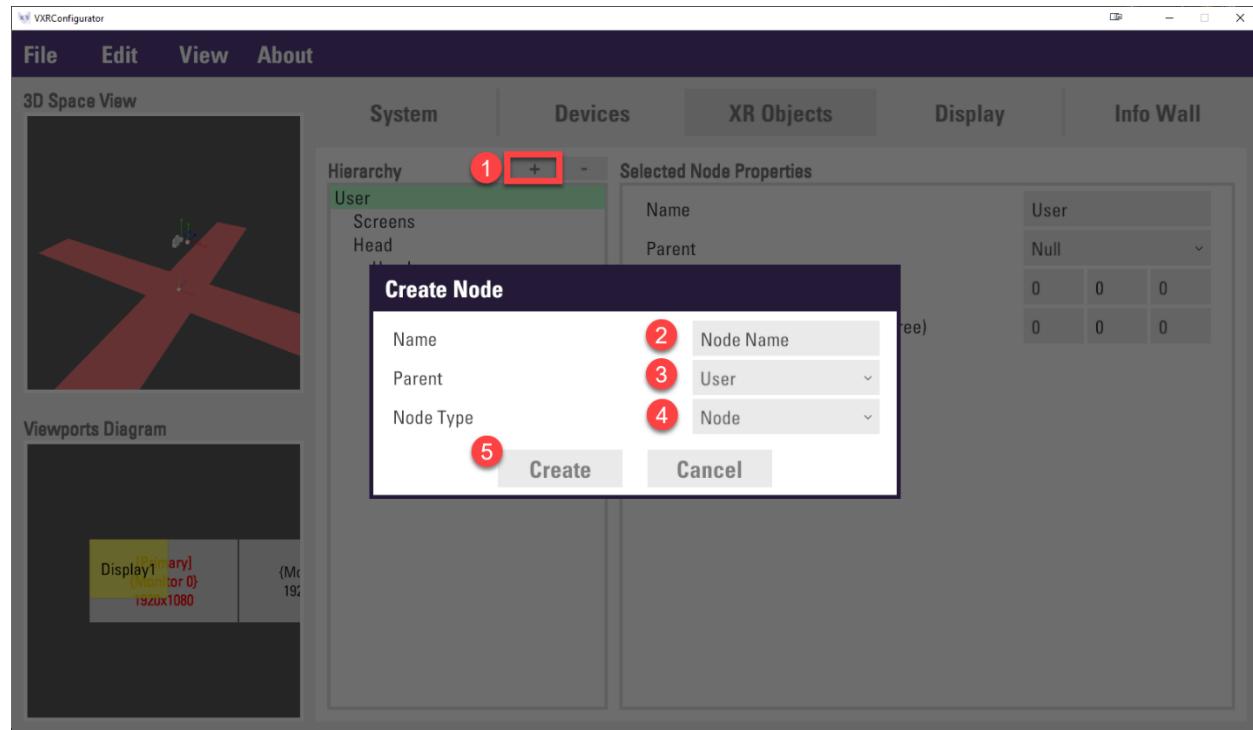
There are 4 built-in system config templates provided in VotanicXR configurator that can be used as a starting point to configuring different types of XR systems. Each type of system config template features a slightly different XR object hierarchy and setting. Below we provide a brief description of the system config templates' XR objects hierarchy:

Config Template	Default Hierarchy	Description
PC	<ul style="list-style-type: none">• User<ul style="list-style-type: none">◦ Screens◦ Head<ul style="list-style-type: none">▪ Hand▪ MainCamera	<p>The default PC config template features a user with a head entity and a hand entity, and a first-person camera attached to the head entity. A virtual tracker is attached to the head entity so that user can use the mouse to rotate the head in runtime.</p> <p>There is no physical screen in a PC config. The image rendered by the main camera is displayed in fixed resolution on the PC's primary display.</p> <p>The PC config is typically used by developer to debug the VotanicXR application on PC monitor using just the mouse and keyboard.</p>
HMD	<ul style="list-style-type: none">• User<ul style="list-style-type: none">◦ Screens<ul style="list-style-type: none">▪ RightScreen◦ Head<ul style="list-style-type: none">▪ MainCamera◦ Hand◦ Hand2• Stabilizer	<p>The default HMD config template features a user with a head entity and two hand entities (Hand and Hand1). The head and hand entities are pre-configured to link to supported SteamVR trackers and controllers.</p> <p>The HMD config features two cameras: the main camera for capturing and showing the default HMD camera image; and a stabilizer camera showing stabilized camera image in the HMD. By default, the stabilizer camera will take priority.</p> <p>The HMD config also includes a screen node which is required for info wall to</p>

		display debug information. The info wall feature can be enabled in-game using the in-game menu.
CAVE	<ul style="list-style-type: none"> • User <ul style="list-style-type: none"> ◦ Screens <ul style="list-style-type: none"> ▪ LeftScreen ▪ FrontScreen ▪ RightScreen ▪ FloorScreen ◦ Head <ul style="list-style-type: none"> ▪ LeftCamera ▪ FrontCamera ▪ RightCamera ▪ FloorCamera ◦ Hand 	<p>The default CAVE config template provides a sample configuration for a 4-sided multiple display system. It features a user with a head and a hand entity, together with 4 screens each assigned to a camera node (4 cameras in total) rendering with perspective view on the display.</p> <p>As the CAVE config template is provided to serve as a starting point for configuring a CAVE system, stereoscopic rendering is turned off by default.</p>
DOME	<ul style="list-style-type: none"> • User <ul style="list-style-type: none"> ◦ Screens ◦ Head <ul style="list-style-type: none"> ▪ Hand ▪ MainCamera 	<p>The default DOME config is similar to the PC config in terms of the structure, except that the DOME config does not have any tracker attached to the head entity.</p> <p>The DOME config is typically used to display the VotanicXR application in immersive DOME system which usually does not feature tracking nor stereoscopic rendering.</p>

Add a New XR Object Node to Hierarchy

By default, the XR Object Hierarchy has a “User” node and a “Screens” node, both of which are serving as structures for grouping different XR objects as its child node. There is also a “Head” and a “Hand” entity in the hierarchy serve to represent the user’s head and hand in the VotanicXR application. A first-person camera is attached to the “Head” entity to capture the VR world viewed from the perspective of the user. Based on the structure and composition of the XR system, one can customize the XR objects hierarchy by configuring the default nodes’ properties and adding new nodes to the hierarchy.

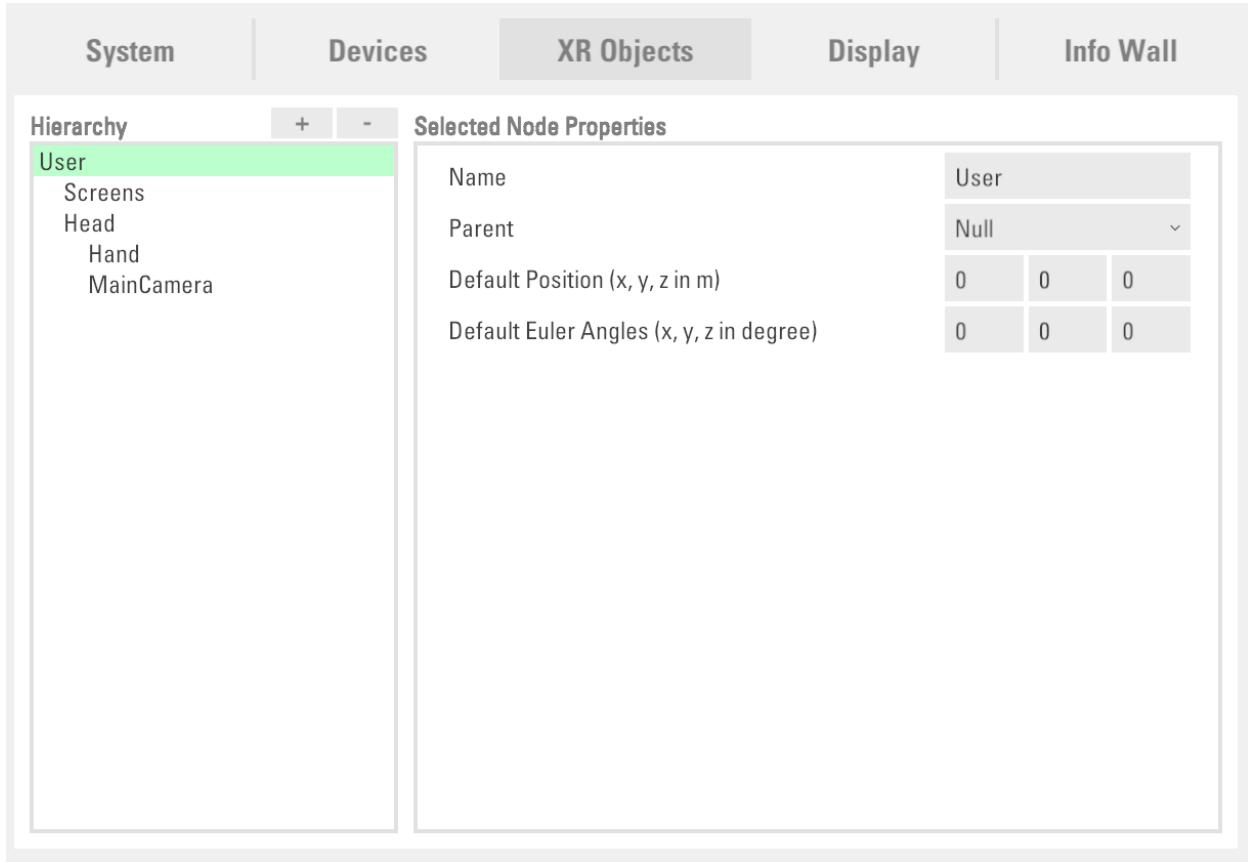


To add a new XR object node, click the “+” button in the Hierarchy view, then in the “Create Node” panel, input the “Name”, and select the “Parent” node and the “Node Type” for the new node. Details of the “Create Node” panel properties are explained below:

Node Property	Description
Name	Self-defined name that can uniquely identify the XR object node. <i>Note: If you add more than one node using the same name, only the first node in the hierarchy will be accessible by VotanicXR.</i>
Parent	A drop-down list for selecting the parent node of the new XR object node. The transform of the node will follow that of the parent node in the VR world.
Node Type	Define the type of the XR object node. The options for a XR object node are: Node , Entity , Screen , and Camera .

Configure Node

The node properties can be configured by first selecting the node in the Hierarchy and then modifying the properties in the “Selected Node Properties” view on the right.



For each node, modify the “Name”, “Parent”, “Default Position” and “Default Euler Angles” (rotation) is necessary. Note that, as the XR objects node follows a hierarchical structure, the default position and rotation for a node refers to the local position and rotation of the node. If the node has a parent node defined, the node will move along with the parent node in the VR world.

Configure Entity

Entity is a special type of Node with Tracking and Input properties. In addition to the basic node properties, it is possible to configure properties related to tracking and input/interaction for an entity node. Details of the Entity node properties are explained below:

Entity Property	Description
Name	Self-defined name that can uniquely identify the XR object node.
Parent	A drop-down list for selecting the parent node of the new XR object node.
Default Position	Default local position of the node when it instantiates in the VR world.
Default Euler Angles	Default local rotation of the node (in Euler angles) when it instantiates in the VR world.
Linked Tracker	<p>Input the name of the tracker to attach it to the entity node. The linked tracker can be:</p> <ul style="list-style-type: none">• Null: for no tracker attached to the entity• Name of a tracker device in the device list <p>The entity node's local position and rotation will follow that of the tracker's when a tracker is successfully linked to the entity. The transform of the entity will be updated in real-time and previewed in the 3D Space View.</p>
Linked Virtual Device	<p>Input the name of the virtual input device to attach it to the entity node. The linked virtual device can be:</p> <ul style="list-style-type: none">• Null: for no input device attached to the entity• Sensor: for enabling the entity to use the Vision Interactor in VotanicXR application• Name of a virtual controller defined in the Unity scene's hierarchy under vGear: for enabling the entity to use the Wand Interactor and the In-Game Tools in VotanicXR application <p>By default, Sensor is set as the linked virtual device for the Head entity, Controller is set as the linked virtual device for the Hand entity, and Controller2 is set as the linked virtual device for the Hand2 entity.</p>
Tracking Mode	<p>Defines how the entity updates its local transform using the data obtained from tracker:</p> <ul style="list-style-type: none">• Both: the entity's local transform will follow the tracker's real-time position and rotation data

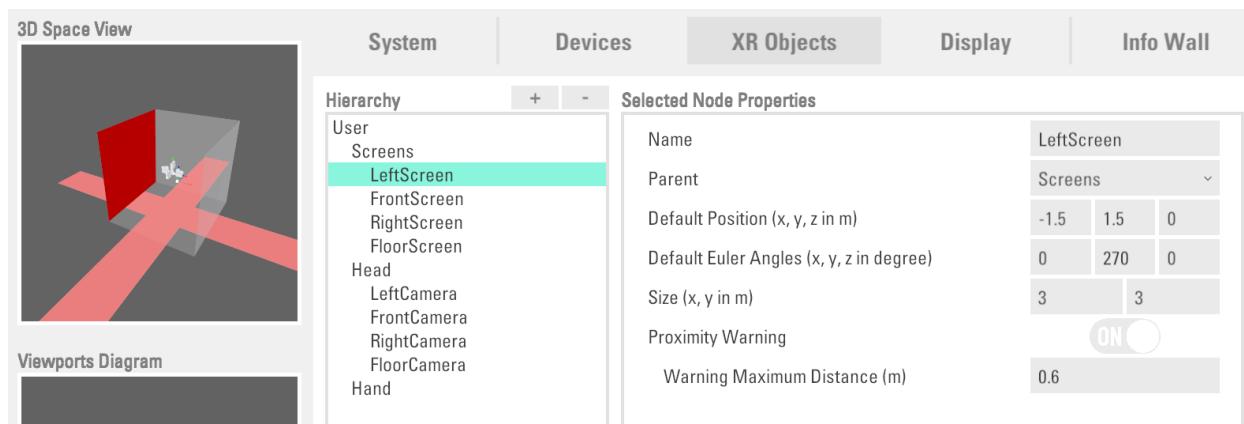
	<ul style="list-style-type: none"> • Position Only: the entity's local transform will follow the tracker's real-time position data only, the default Euler angles or the entity's last-known rotation will be used as the entity's rotation • Rotation Only: the entity's local transform will follow the tracker's real-time rotation data only, the default position or the entity's last-known position will be used as the entity's position • Disable: the entity will not use the tracker data, the default position and default Euler angles or the entity's last-known local transform will be used as the entity's transform
Tracking Position Offset	Defines a fixed position offset value when the linked tracker data is used to update the entity's local transform.
Tracking Euler Angles Offset	Defines a fixed rotation offset value when the linked tracker data is used to update the entity's local transform.
Lost Tracking Action	<p>Defines the action when tracking is lost:</p> <ul style="list-style-type: none"> • Keep Idle: set the local transform of the entity to the last-known tracker transform data • Go to Faraway: move the entity to a faraway location in the VR world so that it's completely off-screen • Reset to Default: reset the transform of the entity to its default local transform • Custom Location: move the entity to a custom location defined below
Lost Tracking Action Specifics	
Retry times	The number of frames for VotanicXR to determine if the tracker device has lost tracking.
Custom Position	Local position for the lost tracking action's custom location.
Custom Euler Angles	Local rotation for the lost tracking action's custom location.

Configure Screen

Screen is a special type of Node with added properties specific to a Screen. “Screen” here refers to rectangular-shaped physical surface in the XR system that is used to display rendered image. In addition to the basic node properties, there are a few more properties specific to the screen node. Details of the Screen node properties are explained below:

Screen Property	Description
Name	Self-defined name that can uniquely identify the XR object node.
Parent	A drop-down list for selecting the parent node of the new XR object node.
Default Position	Default local position of the screen’s center point in the physical world with respect to the origin defined for the XR system (center of the red cross shown in the 3D Space View).
Default Euler Angles	Default local rotation of the screen’s center point in the physical world with respect to the origin defined for the XR system (center of the red cross shown in the 3D Space View).
Size	Physical dimension of the screen.
Proximity Warning	Toggle to switch on / off the proximity warning in VotanicXR application. When proximity warning is on, a warning is displayed when an entity node is close to a screen.
Warning Maximum Distance	Defines the distance between the entity node and the screen for triggering the proximity warning.

The 3D Space View provides some visual reference in configuring the screen properties. When a screen node is created in the hierarchy, a corresponding screen object using the properties defined for the screen will be created in the 3D Space View. The screen object displayed in the 3D Space View will change color when the corresponding screen node is selected in the hierarchy.



Configure Camera

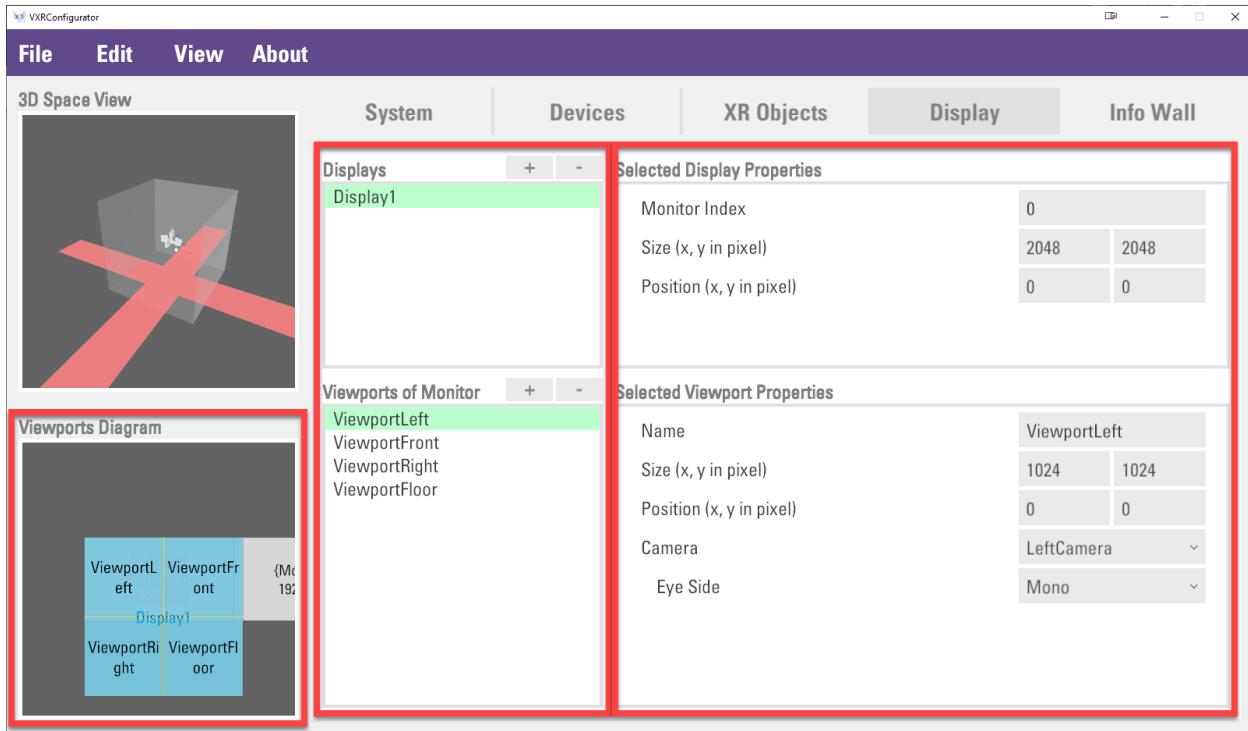
Camera is a special type of Node with added properties related to camera setting. In addition to the basic node properties, it is possible to configure properties related to the camera's behaviour for a camera node. Details of the Camera node properties are explained below:

Camera Property	Description
Name	Self-defined name that can uniquely identify the XR object node.
Parent	A drop-down list for selecting the parent node of the new XR object node.
Default Position	Default local position of the camera node when it instantiates in the VR world.
Default Euler Angles	Default local rotation of the camera node (in Euler angles) when it instantiates in the VR world.
Screen	Choose a screen in the hierarchy to assign to the camera. When a screen is assigned to the camera, the camera will be always facing the screen during runtime and renders in perspective view mode with asymmetric frustum. Otherwise, the camera will render in normal mode with symmetric frustum.
Display Index	Define which display would the camera be rendered to. The display index refers to the index of the display defined in the "Display" tab. The first display defined in the display list has the index "0". Note: <i>If the display index defined in the camera node property here conflicts with the viewport property defined for the camera in the display tab, the display index defined here takes priority. If no viewport is defined for the camera, the camera will render image in full on the specified display.</i>
Is Stereo	Toggle the stereoscopic setting for the camera. If a camera's "Is Stereo" is "ON", two cameras – one for left eye and one for right eye – will be instantiated in runtime, and you can define the camera viewport for the left and right eye camera separately.
Is Inverted Eye	Toggle to swap the position of the left and right eye for a stereo camera. It is only effective if "Is Stereo" is "ON".
Custom Camera View Properties	
Custom Camera View	Toggle the custom camera setting. When Customer Camera View is off, the default camera setting will be used. Note: When a screen is assigned to the Camera, the Field of View and Aspect Ratio cannot be customized.
Field of View	Defines the field of view of the camera.
Aspect Ratio	Defines the aspect ratio of the camera.
Near Clipping Plane	Defines the value of the near clipping plane of the camera.
Far Clipping Plane	Defines the value of the far clipping plane of the camera.

Stabilizer Properties	
Is Stabilizer	<p>Defines whether the camera is a stabilizer camera intended for spectators. It is typically used for generating an extra display showing from the point-of-view of a sync target with built-in stabilizing effect for the spectators to view.</p> <p>When “Is Stabilizer” is “ON”, the camera will behave using the stabilizer properties defined below.</p> <p>Note: <i>For the stabilizer camera to function correctly, The “Parent” and “Screen” property should be set to “Null”, and the “Display Index” should be set to a display intended for the spectators.</i></p>
Enable On Start	Defines whether the stabilizer camera should be enabled at program start.
Sync Target	Defines the name of the node as the stabilizer camera’s sync target. The stabilizer camera will follow the transform of the sync target in runtime.
Position Axes	Defines the axes in which the stabilizer’s position syncs with the position of the sync target.
Rotation Axes	Defines the axes in which the stabilizer’s rotation syncs with the rotation of the sync target.
Lerp Factor	Defines the lerp factor for the stabilizing effect.
Minimum Sync Distance	Defines the minimum distance between the position of the sync target and the position of the stabilizer camera to trigger the stabilizing effect.
Minimum Sync Angle	Defines the minimum angle between the rotation of the sync target and the position of the stabilizer camera to trigger the stabilizing effect.
First Person	<p>Define whether the stabilizer camera is a first-person camera.</p> <p>When “First Person” is “ON”, the “Position Offset” and the “Rotation Offset” properties will be set to 0, so that the camera behave like a first-person camera viewing from the point-of-view of the sync target.</p> <p>When “First Person” is “OFF”, the “Position Offset” and the “Rotation Offset” properties will be set to a preset offset value, so that the camera behave like a third-person camera viewing from a fixed position from the sync target. You can adjust the “Position Offset” and the “Rotation Offset” properties to values based on the requirement of the application.</p>
Position Offset	Defines an offset value for the position of the stabilizer camera with respect to the position of the sync target.
Rotation Offset	Defines an offset value for the rotation of the stabilizer camera with respect to the rotation of the sync target.

Configure Display

The Display tab defines how the images captured by the cameras in VotanicXR will be rendered on the physical monitors of the XR system.



In general, the Display tab can be divided into two views:

- **Display & Viewports View:** defines all the displays in the XR system used to render VotanicXR's application content, and all the camera viewports defined to show on the selected display.
- **Display & Viewports Properties View:** defines the properties of the selected display and viewport.

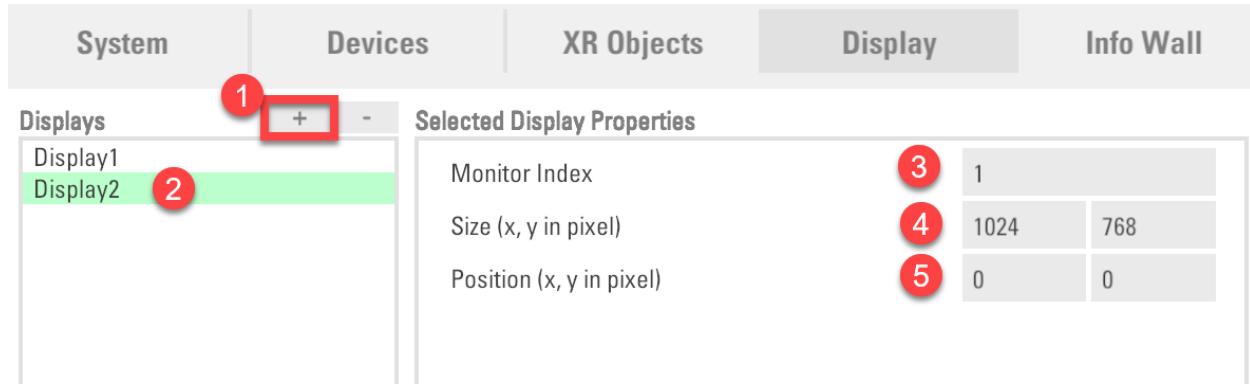
Viewports Diagram

All the displays and viewports will be drawn in the Viewports Diagram on the left-bottom corner. The Viewports Diagram helps to visualize the XR system's physical monitors, the displays, and the viewports, each adds one layer on top of the prior:

Viewport Component	Diagram	Description
Monitor		Shows all the physical monitors connected to the system.
Display		Display is showing on top of the monitor to visualize on what position of a physical monitor will the display be rendered.
Viewport		A viewport is rendered from a single camera source and is drawn on top of a display.

Add and Configure a Display

New display can be added by clicking the “+” button in the display list view. A new display will be created together with a new display drawn in the Viewport Diagram.

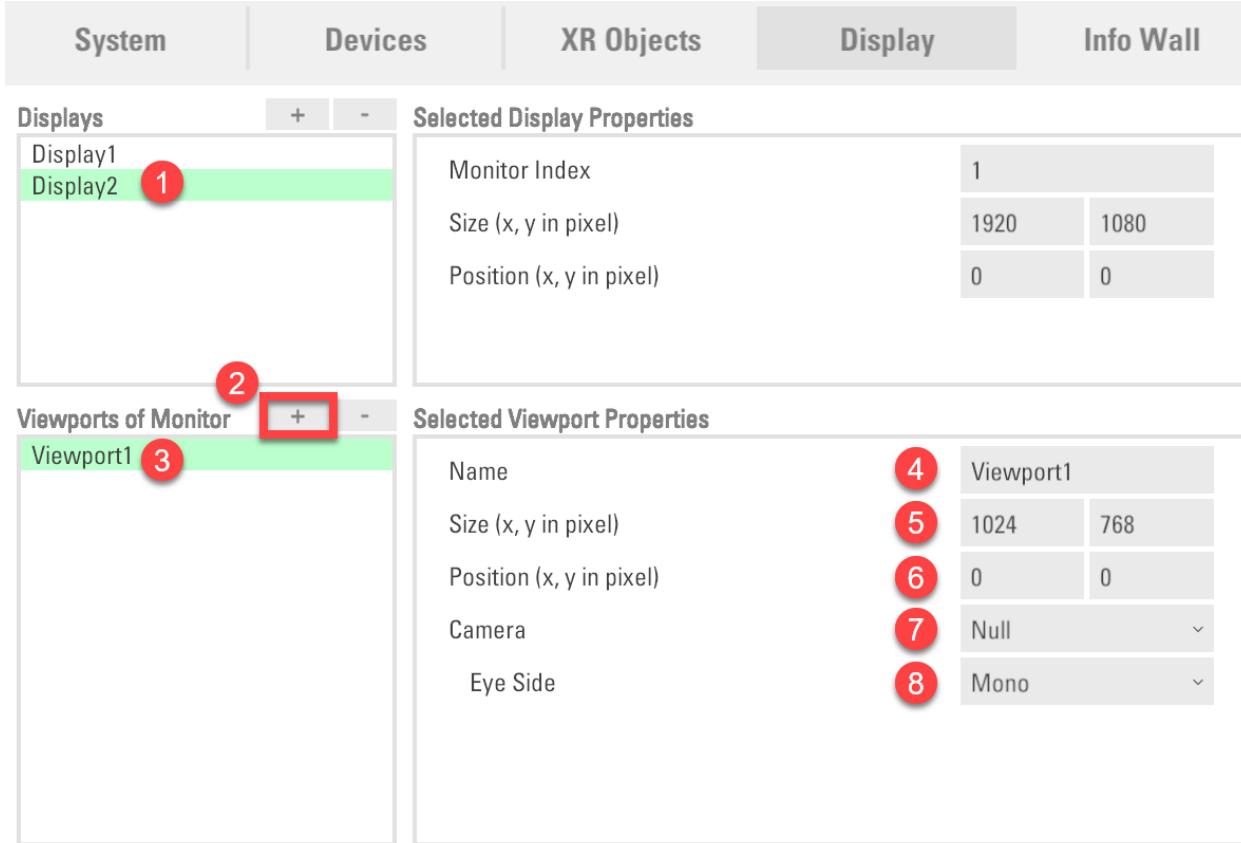


Select the newly created display in the display list, you will be able to configure the details of the display in the Display Properties View. Details of the display properties are explained below:

Display Property	Description
Monitor Index	The PC's physical monitor index in which the display will be shown on.
Size	Size of the display in pixels.
Position	Starting position of the display on the physical monitor. Starting position refers to the where the first pixel of the display is drawn, i.e. the “0, 0” position is the top-left corner.

Add and Configure a Viewport

Viewports can be added to a display by first select the display in the display list and press the “+” button in the Viewports of Monitor list. A new viewport will be created for the display together with a new viewport drawn on the display in the Viewport Diagram.



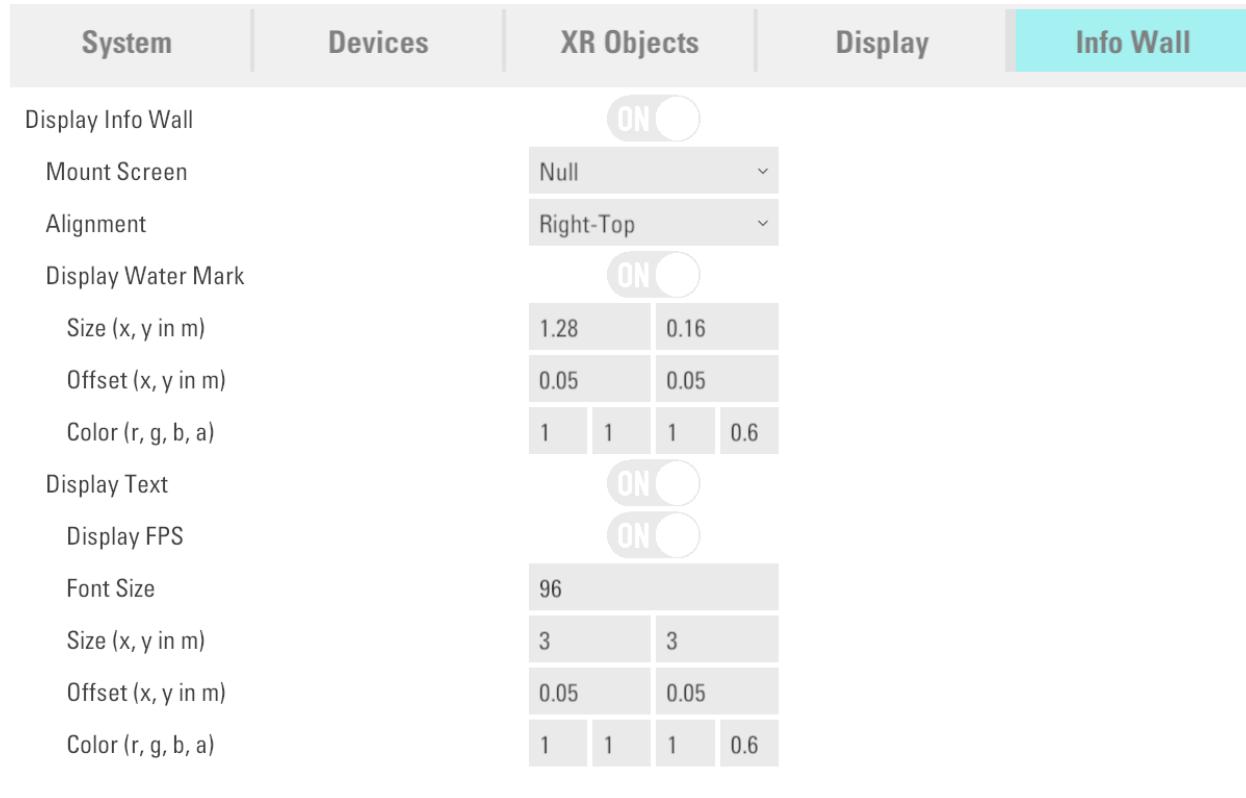
Select the newly created viewport in the Viewports of Monitor list, you will be able to configure the details of the display in the Viewport Properties View. Details of the viewport properties are explained below:

Viewport Property	Description
Name	Self-defined name for the viewport
Size	Size of the viewport in pixels.
Position	Starting position of the viewport on the display. Starting position refers to the where the first pixel of the viewport is drawn, i.e. the position of the viewport's top-left corner.
Camera	A drop-down list for selecting a Camera created in the XR Objects list. The viewport will draw image rendered from the selected Camera.
Eye Side	The target eye-side for the viewport. For a stereo camera, select either the left eye or right eye for the viewport; For a non-stereo camera, the eye side will be set to “Mono” by default.

Note: As a camera in Unity only renders 1 viewport, if more than 1 viewport chooses to render from the same camera with the same eye side, only the viewport that's defined first will be rendered on the display.

Configure Info Wall

The “Info Wall” is used to display information such as the debug log, FPS, etc. in the VotanicXR application. Configuring the Info Wall properties here defines the behaviour of the Info Wall when the application starts. During runtime, these properties can be modified using the [In-Game menu](#) or by calling the VotanicXR [API](#).



Details of the Info Wall properties are described below:

Info Wall Property	Description
Display Info Wall	Toggles on/off to display the info wall object at application start.
Mount Screen	Drop-down list that defines the screen in which the info wall will be displayed. Note: A screen in the XR Objects hierarchy is required to display the info wall.
Alignment	The position on the screen in which the info wall will be displayed.
Display Water Mark	Toggles on/off to display the VotanicXR watermark in the Info Wall.
Water Mark Properties	
Size	Defines size of the water mark.

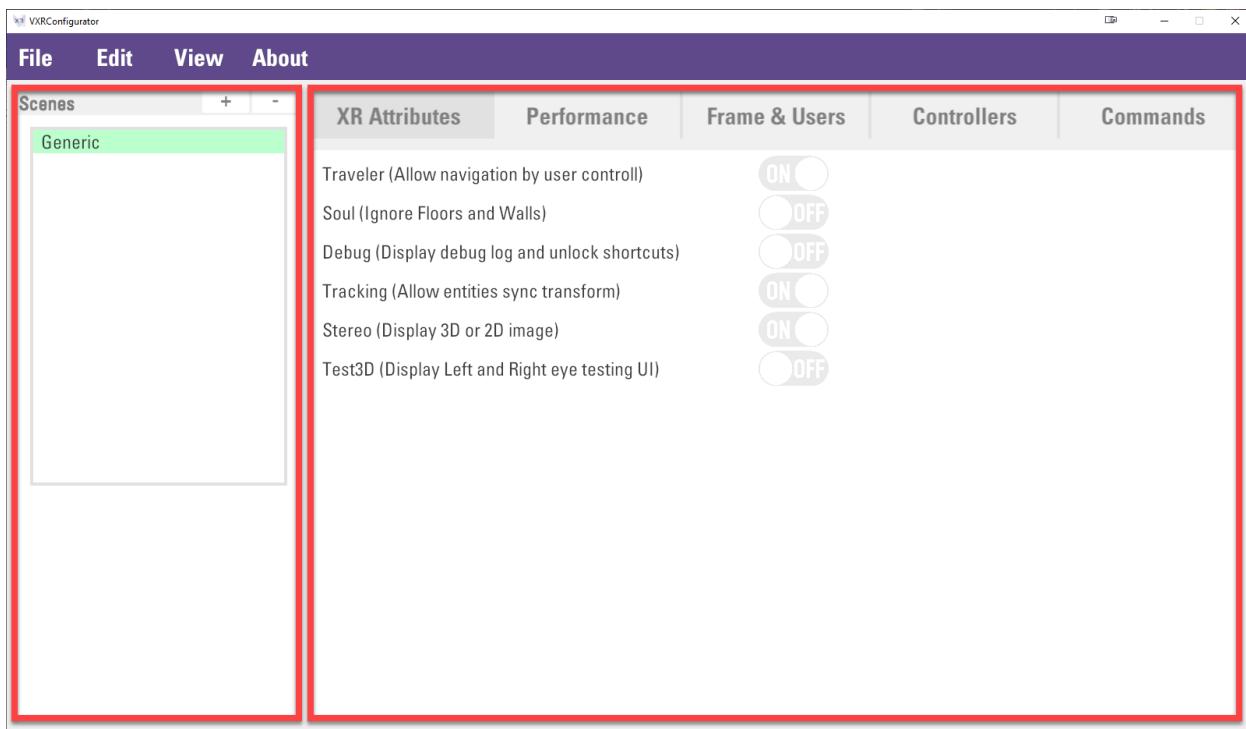
Offset	Defines the offset value of the water mark from the chosen alignment position.
Color	Defines the color and alpha value of the water mark.
Display Text Properties	
Display Text	Toggles on/off to display all text info in the Info Wall.
Display FPS	Toggles on/off to display FPS info in the Info Wall.
Font Size	Defines the font size of the text shown in the Info Wall.
Size	Defines the size of the area for displaying text.
Offset	Defines the offset value of the text area in the Info Wall.
Color	Defines the color and alpha value of the text.

Application Settings

The application settings files, with file extension “.vxrs”, defines the specific settings of a VotanicXR application. Properties in the application settings file are responsible for how an application uses VotanicXR, such as the XR Attributes, the input and commands settings, etc.

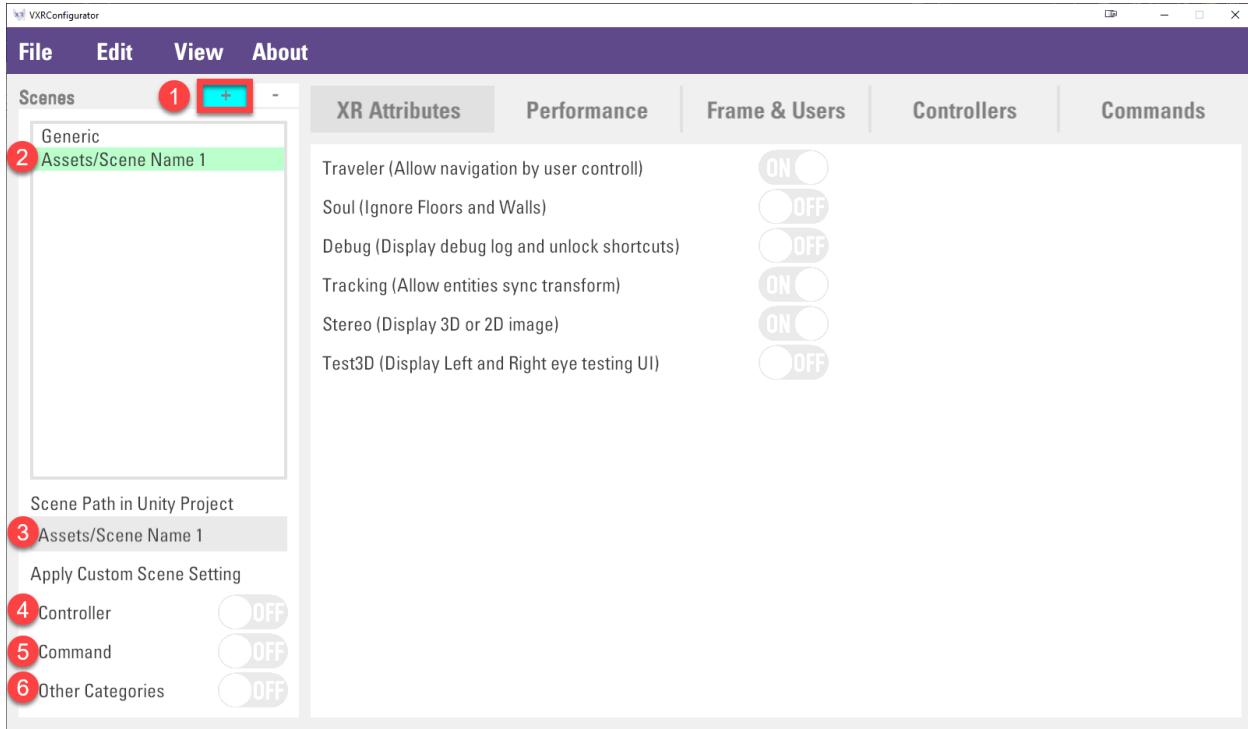
Application Settings Overview

The application settings configuration view in the VotanicXR Configurator is mainly divided into two segments. On the left-hand side is the “Scenes” setting, which specifies the name of the scene in the VotanicXR application in which those settings on the right-hand side apply to. The large area on the right-hand side stores different properties of the application setting, separated into different groups for ease of accessing different properties.



Add a Scene Profile

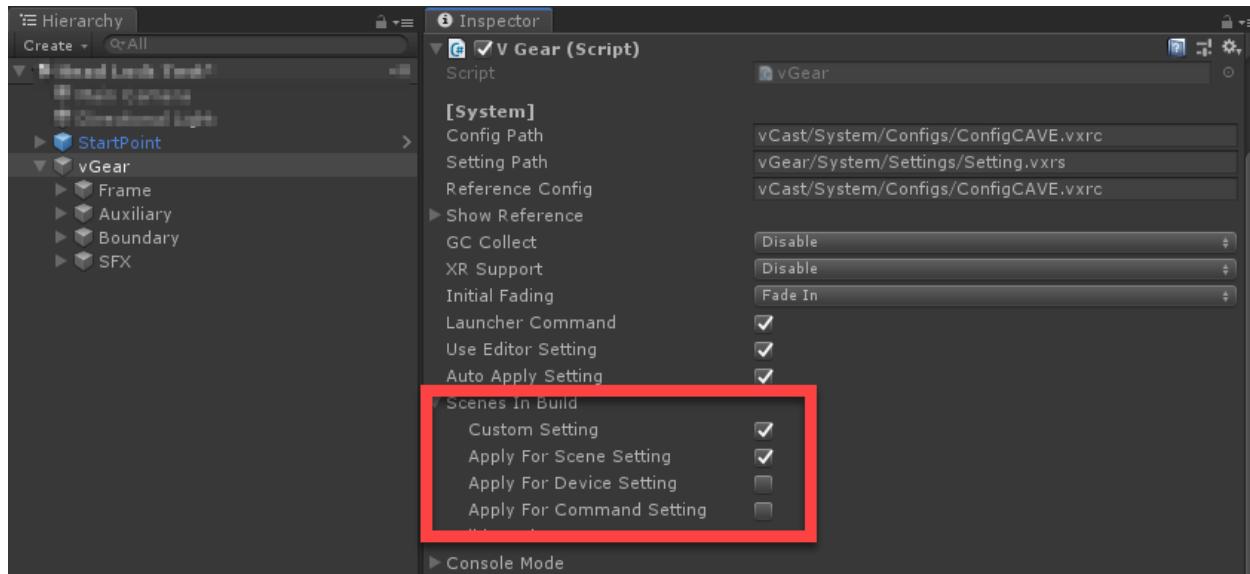
To apply different application settings in different scenes of the VotanicXR application, a new Scene Profile can be added in the application setting to configure different properties for the specific scene.



Select the “+” button in the scene list view, a new Scene Profile will be created in the scene list. Select the newly created scene profile to configure the specifics:

Scene Profile Property	Description
Scene Path in Unity Project	Defines the path of the scene in the Unity project.
Apply Custom Scene Setting in	
Controller	Toggle to apply the custom virtual controller setting to the scene.
Command	Toggle to apply the custom commands setting to the scene.
Other Categories	Toggle to apply other custom settings to the scene.

Note: it is necessary to tick the “Custom Setting” in the scene’s vGear setting in Unity in order to use the custom scene profile. Otherwise, the settings defined in the “Generic” scene profile will be used.



Configure XR Attributes

The XR Attributes define the default XR behaviour of the scene when it is first loaded.

XR Attributes	Performance	Frame & Users	Controllers	Commands
Traveler (Allow navigation by user control)		<input checked="" type="radio"/> ON		
Soul (Ignore Floors and Walls)		<input type="radio"/> OFF		
Debug (Display debug log and unlock shortcuts)		<input type="radio"/> OFF		
Tracking (Allow entities sync transform)		<input checked="" type="radio"/> ON		
Stereo (Display 3D or 2D image)		<input checked="" type="radio"/> ON		
Test3D (Display Left and Right eye testing UI)		<input type="radio"/> OFF		

Under the XR Attributes tab, 6 key XR attributes can be defined:

XR Attribute	Description
Traveler	Toggles the setting to allow user to navigate in the VR world using movement commands. Details of the movement commands are explained in the Locomotion tutorial .
Soul	Toggles the setting of the user transform behaviour in locomotion regarding gravity and collision. Details of the soul mode is explained in the Locomotion tutorial's movement modes section .
Debug	Toggles the setting to enable debug mode when the scene starts. Details of the debug mode, can be found in the debug mode documentation.
Tracking	Toggles the setting to enable tracking when the scene starts.
Stereo	Toggles the setting to start the scene in 3D or in 2D mode.
Test3D	Toggles the setting to start the scene in Test 3D mode.

Configure Performance Properties

The performance properties are settings relate to the quality of the VotanicXR application, they can be configured under the Performance tab.

XR Attributes	Performance	Frame & Users	Controllers	Commands
Refresh Rate	0.2			
Target Frame Rate	120			
Force Quality	<input checked="" type="checkbox"/> ON			
Level	0			
VSync	Don't Sync			
Shadow	Hard			
Resolution	High			
Projection	Close Fit			
Distance (m)	100			
Near Plane Offset	1			
Light Count	1			
Anisotropic	Forced On			
Anti Aliasing	8x			

Details of the performance properties are as follows:

Performance Property	Description
Refresh Rate	Defines the rate in which VotanicXR's features checks the application. Default value is 0.2 second, reduce the value may impact the performance.
Target Frame Rate	Defines the target frame rate of the application.
Force Quality	Toggles the setting to force the scene to run under a certain quality setting.
Level	Defines the quality level if the "Force Quality" setting is ON. This value is referring to Unity quality level profile. The list of quality levels be checked in Unity "Edit -> Project Settings -> Quality". The topmost level profile is index 0.
VSync	Defines the VSync setting in the application's quality setting. The available options are: <ul style="list-style-type: none">• Don't Sync

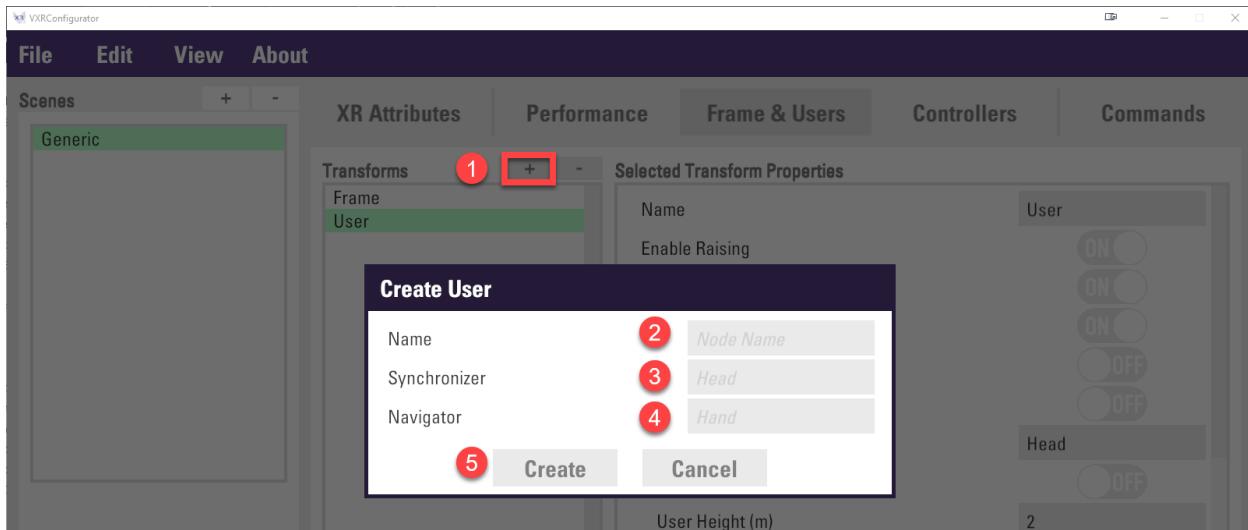
	<ul style="list-style-type: none"> • Every V Blank • Every 2 V Blank
Shadow Setting	
Shadow	<p>Defines the shadow setting of the application's quality setting. The available options are:</p> <ul style="list-style-type: none"> • Disable • Hard • Hard and Soft
Resolution	<p>Defines the resolution in the shadow setting. The available options are:</p> <ul style="list-style-type: none"> • Low • Medium • High • Very High
Projection	<p>Defines the projection in the shadow setting. The available options are:</p> <ul style="list-style-type: none"> • Close Fit • Stable Fit
Distance	Defines the distance in the shadow setting.
Near Plane Offset	Defines the near plane offset in the shadow setting.
Light Count	Defines the light count in the quality setting.
Anisotropic	<p>Defines the anisotropic in the quality setting. The available options are:</p> <ul style="list-style-type: none"> • Disable • Per Texture • Forced On
Anti Aliasing	<p>Defines the anti-aliasing in the quality setting. The available options are:</p> <ul style="list-style-type: none"> • Disable • 2x • 4x • 8x

Configure Frame & User

The Frame & User properties define various properties of the Frame and User GameObject in vGear. In VotanicXR, Frame and User are XR Object transforms which can be configured based on the needs of the VotanicXR application.

Add a New User

For typical use, the VotanicXR application would normally have only 1 User. In case the application has more than 1 user, there is more than 1 “User” node in the XR Objects hierarchy in the system configuration file, and in the Unity scene’s hierarchy under vGear. To configure the other user in the VotanicXR Configurator, it is necessary to add a new User in the Frame & User setting.

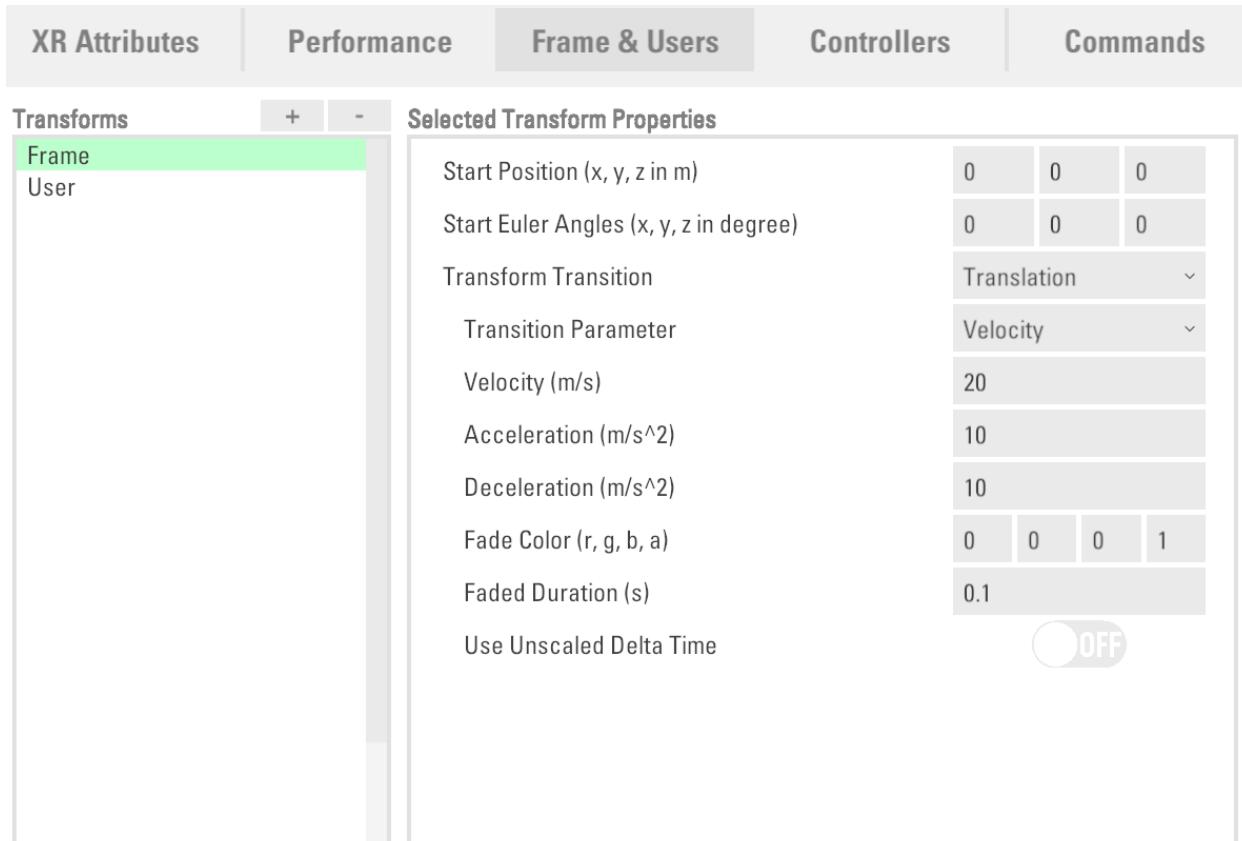


To add a new User node, select the “+” button on the Frame & User transforms list, then input the name, synchronizer, and navigator node for the user and press “Create”. Details of the User properties are as follows:

User Property	Description
Name	Name of the User node in the XR Objects hierarchy.
Synchronizer	Name of the User’s head entity in the XR Objects hierarchy for the specific user. It will be used to determine the transform of the User’s character controller in the application.
Navigator	Name of the User’s hand entity in the XR Objects hierarchy for the specific user. It will be used to determine the user’s forward direction in the application.

Configure Frame Transform

The Frame Transform properties define the properties of the Frame GameObject under vGear in the Unity scene's hierarchy. As there is only 1 Frame object in a VotanicXR application, it is not possible to create additional Frame transform in the Transforms list.



To configure the Frame transform properties, select the “Frame” in the Transforms list and edit the Transform Properties on the right panel. Details of the Frame Transform properties are explained below:

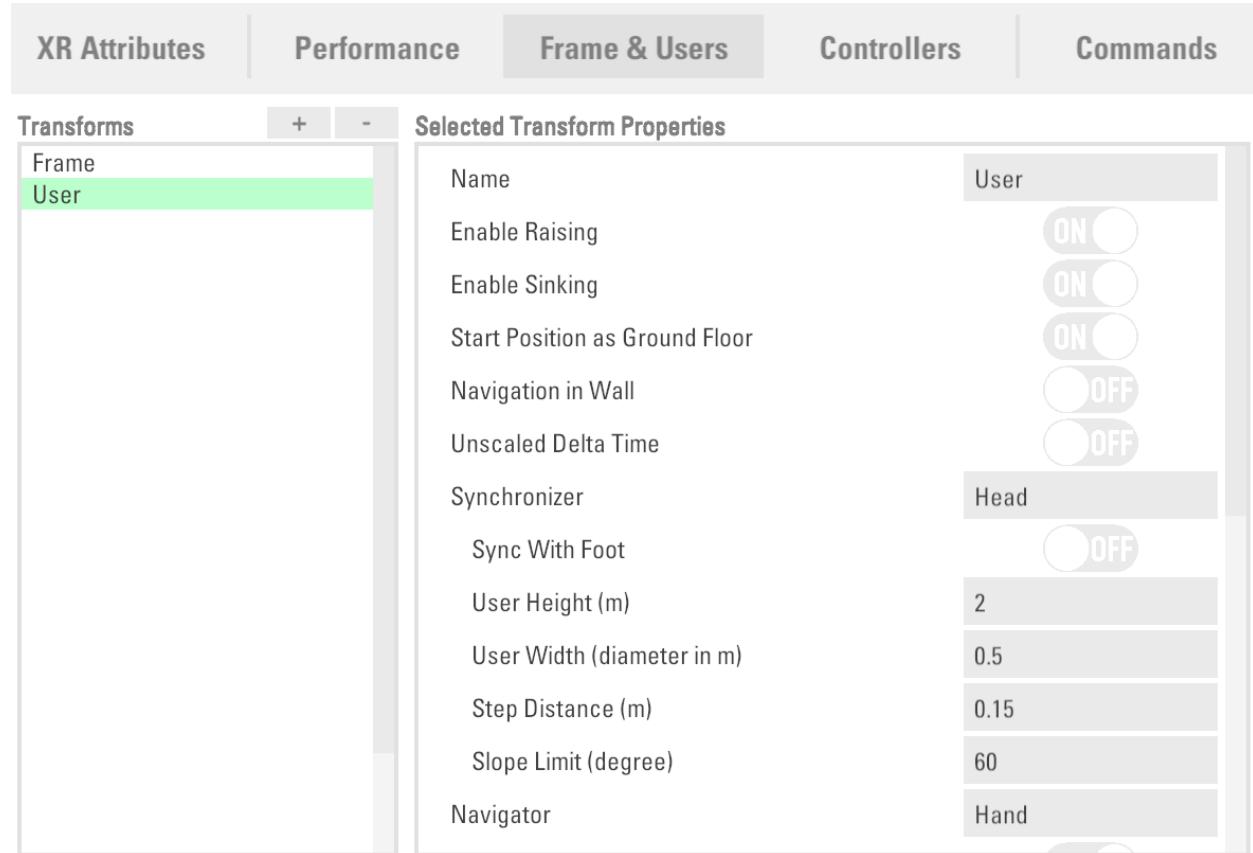
Frame Property	Transform Description
Start Position	Defines the position of the Frame transform when the scene starts. If you have set the StartPoint object in the scene's hierarchy, this value will be set automatically using the StartPoint's position when you build the application.
Start Euler Angles	Defines the rotation of the Frame transform when the scene starts. If you have set the StartPoint object in the scene's hierarchy, this value will be set automatically using the StartPoint's rotation when you build the application.

Transform Transition Properties	
Transform Transition	<p>Defines the transform transition effect mode when you call the Transform API to move the Frame transform. The available transition effect modes are:</p> <ul style="list-style-type: none"> • Fade • Translation
Transition Parameter	<p>Defines the transform mode when you call the Transform API to move the Frame transform. The available transform transition modes are:</p> <ul style="list-style-type: none"> • Velocity: the velocity-based transition properties will be used when moving the Frame transform. The transition will start with the specified Acceleration value until the Frame object's movement velocity reaches the specified Velocity, it will then transition with the specified Velocity until it's close to the destination, then the Frame transform transition will slow down with the specified Deceleration value and finally comes to a stop at the destination position. • Time: the time-based transition properties will be used when moving the Frame transform. The Frame object will first accelerate for the specified Acceleration Duration, moves in uniform speed for the specified Uniform Speed Duration, and then slow down for the Deceleration Duration until finally comes to a stop at the destination position.
Velocity-based Transition Properties	
Velocity	<p>Defines the targeted velocity of the Frame transform object for the velocity-based transition.</p> <p>Note: if the Frame transform's destination is too close to its current position, the transition process may not be able to reach the targeted velocity.</p>
Acceleration	Defines the Frame transform object's acceleration value for the velocity-based transition.
Deceleration	Defines the Frame transform object's deceleration value for the velocity-based transition.
Time-based Transition Properties	
Uniform Speed Duration	Defines the duration in which the Frame transform object will move in constant velocity.
Acceleration Duration	Defines the duration in which the Frame transform object accelerates at the start of the transition.

Deceleration Duration	Defines the duration in which the Frame transform object decelerates at the end of the transition.
Fade Color	Defines the color the screen rendering will fade to when the Fade transition effect is used.
Faded Duration	Defines the duration of the fade transition.
Use Unscaled Delta Time	Toggles the option to use unscaled delta time during transform transition. When this option is ON, the time scale defined in the application will be used.

Configure User Transform

The User Transform properties define the properties of the selected User GameObject under vGear's Frame GameObject in the Unity scene's hierarchy.



To configure the User transform properties, select the specific User in the Transforms list and edit the User Properties on the right panel. Details of the User Transform properties are explained below:

User Property	Transform	Description
Name		Defines the name of the User GameObject in the Unity scene's hierarchy.
Enable Raising		Toggles the option to enable elevating the User transform to the upper floor level when the User transform's Synchronizer is above the collider of the upper floor level.
Enable Sinking		Toggles the option to enable sinking the User transform to the below floor level when the User transform's Synchronizer is below the collider of the current floor level.

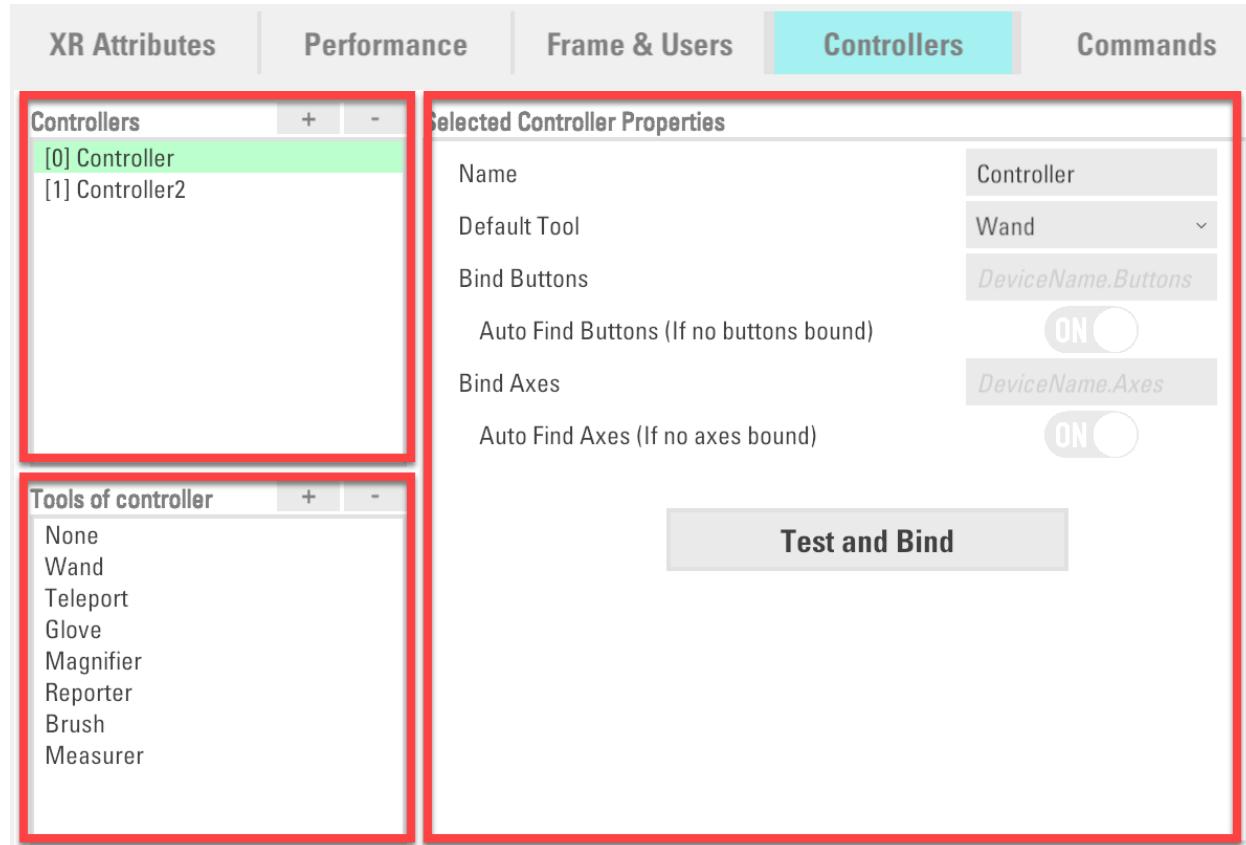
Start Position as Ground Floor	Toggles the option to automatically set the level of the StartPosition GameObject as Ground Floor if no floor is set.
Navigation in Wall	Toggle to allow the user moves through the wall.
Unscaled Delta Time	Toggle to use unscaled delta time in the User transform. The time of the User Character will not be affected by the global time scale in Unity.
Synchronizer Properties	
Synchronizer	Defines the name of the head Entity used to determine the Center of the User's Character Controller. It is used for determining the User's Character position.
Sync With Foot	<p>Toggles the option to also use the Foot Entities to determine the Center of the User's Character Controller.</p> <p>Note: the default names for the Foot Entities are “FootL” and “FootR” in the XR Objects’ hierarchy. For the second user or after, the default names will be “FootL2” and “FootR2” and so on. It is necessary to set up the entities in the hierarchy and link a tracker for both of them, in order to use this feature effectively.</p>
User Height	Defines the height of the Character’s Capsule Collider.
User Width	Defines the width of the Character’s Capsule Collider, i.e. the length of the Capsule Collider’s radius.
Step Distance	Defines the step offset of the Character Controller.
Slope Limit	Defines the Slope Limit of the Character Controller.
Navigator Properties	
Navigator	Defines the name of the hand entity used to determine the User’s forward direction during navigation.
Use Two Hands	<p>Toggles to enable using both hands’ direction to determine the forward direction during navigation.</p> <p>Note: the default name for the second-hand entity is “Hand2” in the XR Objects’ hierarchy. For the second user or after, the default name of second hand will be “Hand2”, Hand4” and so on. It is necessary to set up the entity in the hierarchy and link a tracker to use this feature properly.</p>
Move Speed	Defines the movement speed of the User when using the movement commands.
Rotate Speed	Defines the rotation speed of the User during movement.
Jump Height	Defines the height the User can jump when using the JumpSink command.
Acceleration	Defines the acceleration value when using the Move, Rotate, and JumpSink commands.
Transform Transition Properties	

Transform Transition	Defines the transform transition effect mode when you call the Transform API or use the Teleport tool to move the User transform. The available transition effect modes are: <ul style="list-style-type: none"> • Fade • Translation
Transition Parameter	Defines the transform mode when you call the Transform API or use the Teleport tool to move the User transform. The available transform transition modes are: <ul style="list-style-type: none"> • Velocity: the velocity-based transition properties will be used when moving the User transform. The transition will start with the specified Acceleration value until the User object's movement velocity reaches the specified Velocity, it will then transition with the specified Velocity until it's close to the destination, then the User transform transition will slow down with the specified Deceleration value and finally comes to a stop at the destination position. • Time: the time-based transition properties will be used when moving the User transform. The User object will first accelerate for the specified Acceleration Duration, moves in uniform speed for the specified Uniform Speed Duration, and then slow down for the Deceleration Duration until finally comes to a stop at the destination position.
Velocity-based Transition Properties	
Velocity	Defines the targeted velocity of the User transform object for the velocity-based transition. Note: if the User transform's destination is too close to its current position, the transition process may not be able to reach the targeted velocity.
Acceleration	Defines the User transform object's acceleration value for the velocity-based transition.
Deceleration	Defines the User transform object's deceleration value for the velocity-based transition.
Time-based Transition Properties	
Uniform Speed Duration	Defines the duration in which the User transform object will move in constant velocity.
Acceleration Duration	Defines the duration in which the User transform object accelerates at the start of the transition.
Deceleration Duration	Defines the duration in which the User transform object decelerates at the end of the transition.

Fade Color	Defines the color the screen rendering will fade to when the Fade transition effect is used.
Faded Duration	Defines the duration of the fade transition.
Use Unscaled Delta Time	Toggles the option to use unscaled delta time during transform transition. When this option is ON, the time scale defined in the application will be used.

Configure Controllers

The Controllers tab provides an interface to configure the virtual controller used in VotanicXR application. As covered in the [Cross-Platform Input System tutorial](#), the virtual controller is VotanicXR's physical input abstraction layer that unifies input events from various physical devices. It provides interfaces for developers or other VotanicXR modules to access the events of the physical input devices through the virtual controller.

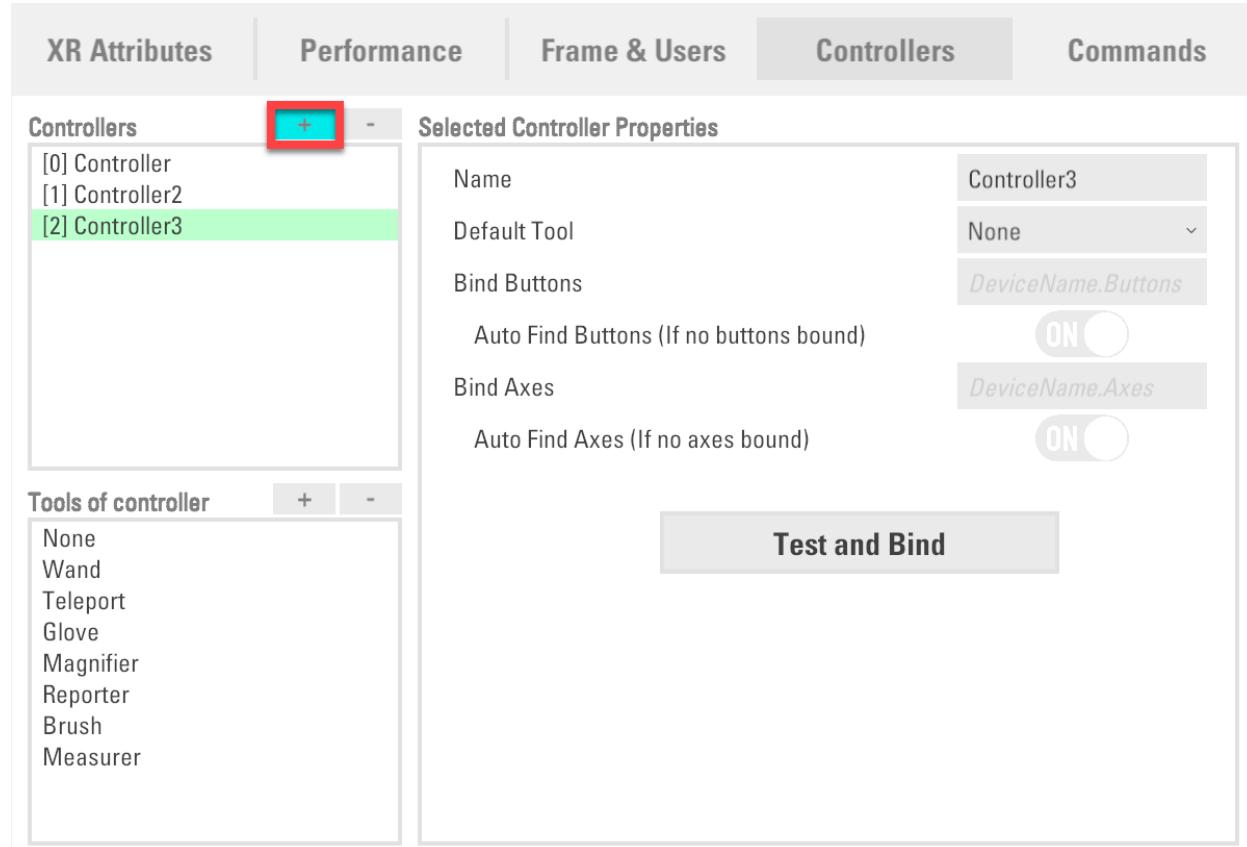


In general, the Controllers tab can be divided into three views:

- **Controllers View:** defines all the virtual controllers for use by the VotanicXR application.
- **Tools of Controller View:** defines all the tools available to use by the selected controller.
- **Controller / Tools Properties View:** defines the properties of the selected controller or tool.

Add a New Controller

To add a new controller, select the “+” button at the Controllers view, a new controller will be added to the controllers list. By default, all of VotanicXR’s in-game tools are attached to the controller, if certain tools are not needed on the controller, you can select the tool in the Tools of Controller view and delete it.



The controllers in the Controllers view are displayed in the format “[i] Controller Name”, where “i” is the controller index to be used in the VotanicXR application, especially in configuring the Commands.

Configure a Controller

Virtual Controller configuration is discussed in detail in the [Cross-Platform Input System tutorial](#), please refer to the guides in the tutorial on the configuration of virtual controller.

Configure Controller Tools

Properties of a tool on a controller can be configured by first selecting the controller in the Controllers view and selecting the respective tool in the Tools of Controller view. Each tool can be configured individually at the Selected Controller Properties view on the right. Details of the in-game tools are explained in the [Using In-Game Tools tutorial](#). The tools available for configuration are described below:

Tool	Description
None	Configures the behaviour of the controller when “None” is set as the default tool.
Wand	Configures the behaviour of the controller when the Wand tool is used.
Teleport	Configures the behaviour of the controller when the Teleport tool is used.
Glove	Configures the behaviour of the controller when the Glove tool is used.
Magnifier	Configures the behaviour of the controller when the Magnifier tool is used.
Reporter	Configures the behaviour of the controller when the Reporter tool is used.
Brush	Configures the behaviour of the controller when the Brush tool is used.
Measurer	Configures the behaviour of the controller when the Measurer tool is used.

When a tool is selected, the tool properties can be configured on the right panel. Details of the tool properties are explained below:

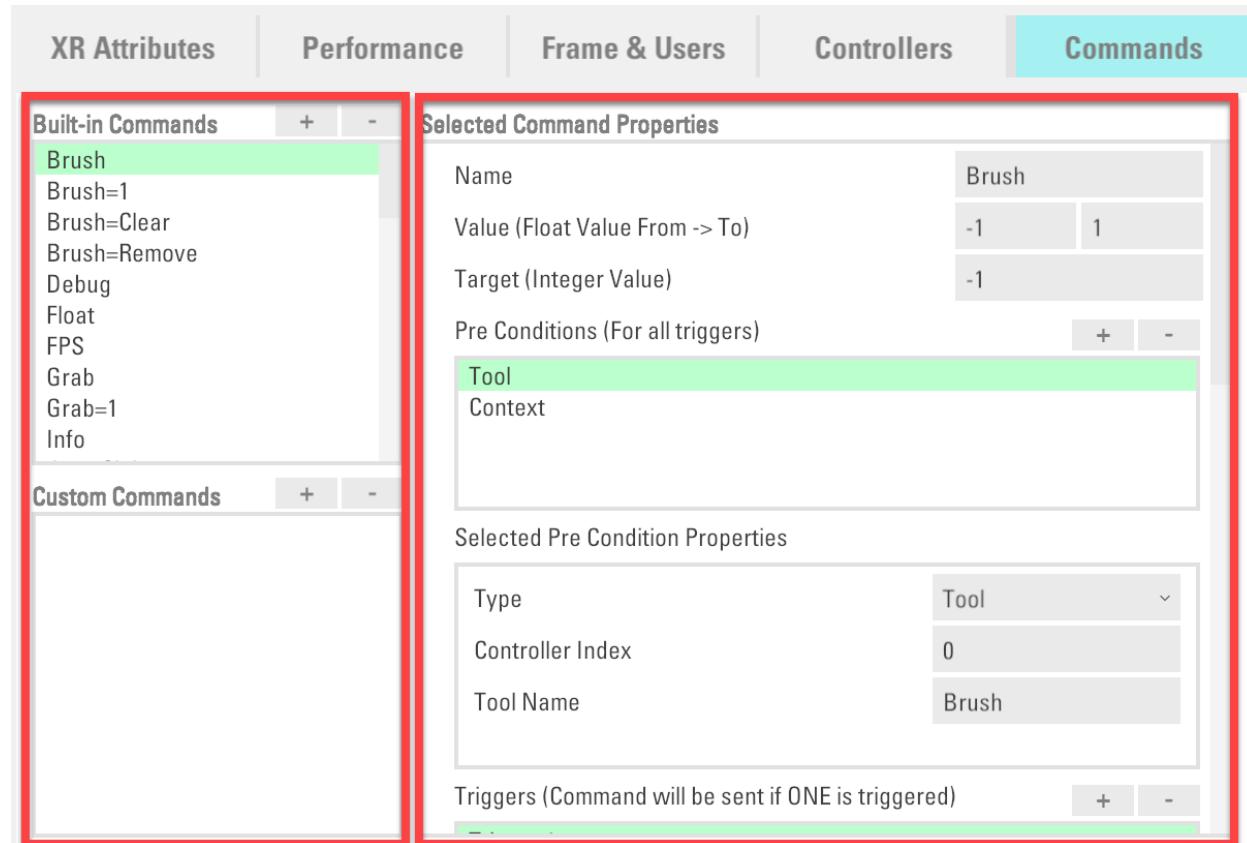
Tool Property	Description
Auto Switch to other tool	Toggles to allow this tool to switch to other tool when “Enable Scan” condition is fulfilled.
Enable Scan and switch to this tool	Toggles to allow scanning and automatically switch into this tool.
Scan Length	Defines the Raycast length for tool detection.
Raycast with priority	Toggles to enable raycast with priority. If the priority option is ON, the raycast will prioritize the raycast layer according to the layer order defined in the Raycast layers property. If the priority option is OFF, Unity’s default raycast behaviour will be used.
Raycast layers	Defines the layers in which the tool’s raycast will have effects on.
Ignore layers	Defines the layers that will be ignored by the tool’s raycast.
Position Offset	Defines the position offset of the tool with respect to the position of the entity (in XR Objects) which the controller is attached to.
Euler Angles Offset	Defines the rotation offset of the tool with respect to the rotation of the entity (in XR Objects) which the controller is attached to.
Wand Ray Interactive Properties	

Wand Ray Interactive	Toggles to enable interaction using the Wand Ray together with the tool when this tool is active.
Disable	Toggles to set the Wand Ray to disable state.
Display Ray	Toggles to enable displaying the Wand Ray.
Render Queue	Defines the render queue of the Wand Ray.
Always On Top	Toggle to set the Wand Ray to be displayed always on top.
Normal State Color	Defines the color of the Wand Ray when it's at normal state.
On Top Alpha	Defines the alpha value of the Wand Ray when it is displayed on top at normal state.
Select State Color	Defines the color of the Wand Ray when it's at select state.
On Top Alpha	Defines the alpha value of the Wand Ray when it is displayed on top at select state.
Press State Color	Defines the color of the Wand Ray when it's at press state.
On Top Alpha	Defines the alpha value of the Wand Ray when it is displayed on top at press state.
Disable State Color	Defines the color of the Wand Ray when it's at disable state.
On Top Alpha	Defines the alpha value of the Wand Ray when it is displayed on top at disable state.
Glove Interactive Properties	
Glove Interactive	Toggles to enable interaction using the Glove together with the tool when this tool is active.
Display Glove	Toggles to enable displaying the Glove.
Detect Radius	Defines the detection radius of the Glove to be used as range in Unity's physics calculation.
Model Position Offset	Defines the position offset of the Glove model when used together with the active tool.
Normal State Outline	Toggles to enable display an outline on the Glove model when the Glove is at normal state.
Color	Defines the color of the outline when the Glove is at normal state.
Width	Defines the width of the outline when the Glove is at normal state.
Select State Outline	Toggles to enable display an outline on the Glove model when the Glove is at select state.
Color	Defines the color of the outline when the Glove is at select state.
Width	Defines the width of the outline when the Glove is at select state.
Press State Outline	Toggles to enable display an outline on the Glove model when the Glove is at select state.
Color	Defines the color of the outline when the Glove is at press state.
Width	Defines the width of the outline when the Glove is at press state.

Details of the tool properties here complement the guide on configuring in-game tools in the [tutorial](#).

Configure Commands

The Commands tab provides an interface to configure the commands used in VotanicXR applications. The main concepts of the commands are introduced in the Cross-Platform Input System tutorial, where the commands decouple the application logic and the user's physical input device events.

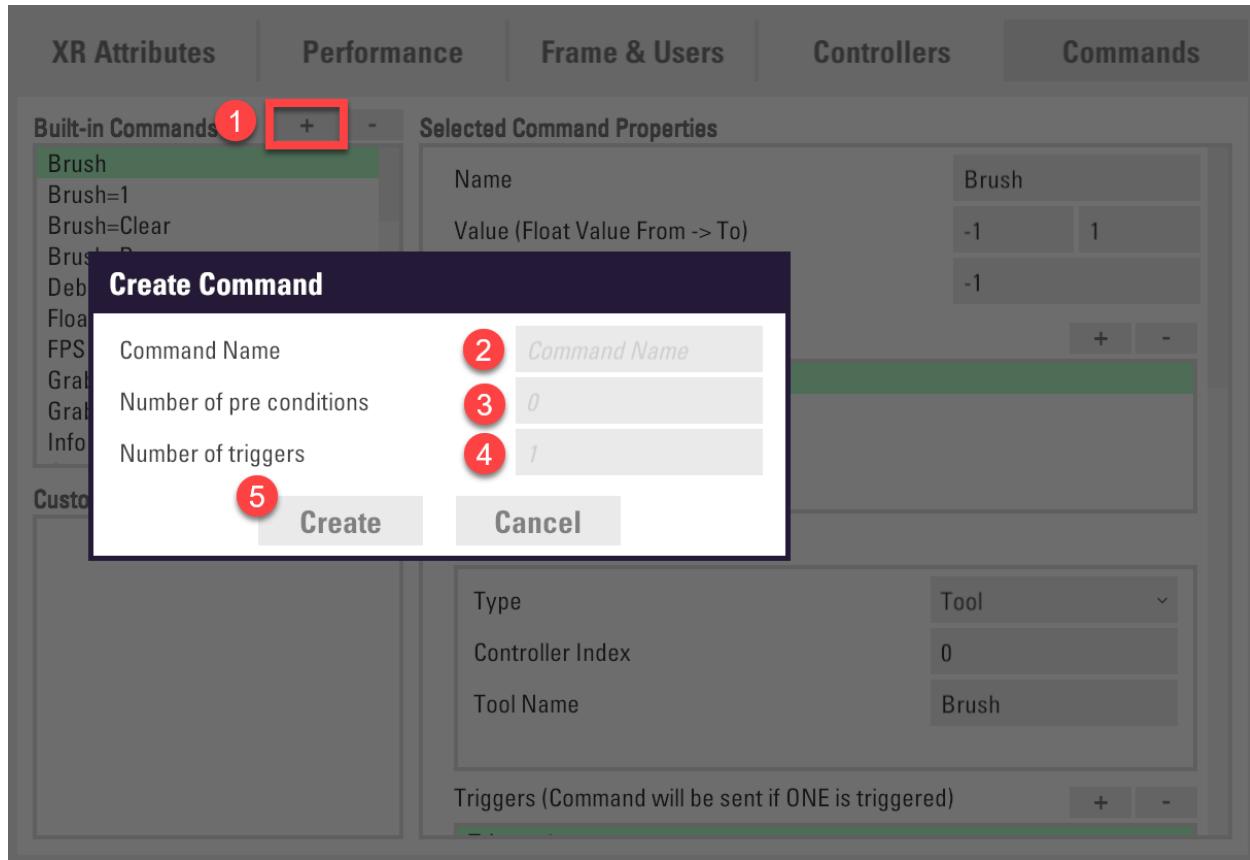


In general, the Commands tab can be divided into 2 views:

- **Commands View:** defines all build-in and custom commands for use by the VotanicXR application.
- **Command Properties View:** defines the properties of the selected command.

Add a New Command

New command can be added by selecting the “+” button at the Built-in Commands list or the Custom Commands list.



At the Create Command panel, specify the command name, number of pre-conditions, and number of triggers then select the “Create” button. Details of the Create Command properties are explained below:

Selected Property	Command Description
Command Name	Self-defined name of the command. For adding a built-in command, the command name shall be selected from a dropdown list of pre-set command names.
Number of pre conditions	Defines the number of pre conditions for the command. The minimum number of pre conditions for a command is 0.
Number of triggers	Defines the number of triggers for the command. The minimum number of triggers for a command is 0.

Configure Command Properties

The command properties can be configured by selecting the command in the commands list and then specifying the details in the Command Properties view on the right. Details of the command properties are explained below:

Command Property	Description
Command Name	Self-defined name of the command.
Value	Defines the range of returning value when the Cmd.Value API is called. By default, the command value has a range -1 to 1, which matches the virtual controller's return value. You can define a customized return value range here if there is a need to scale up or down the return value in the VotanicXR application.
Target	Defines an integer value to be sent out when the command is triggered. This value can be received via the API Cmd.Target.
Pre Conditions	Defines the list of pre-conditions the command must all fulfill in order to trigger the command.
Triggers	Defines the list of triggers in which fulfilling any of the trigger conditions can send out the command when all pre-conditions are met.

Pre conditions can be added by selecting the “+” button at the Pre Conditions list section.

Selected Command Properties

The screenshot shows the 'Selected Command Properties' interface. At the top, there is a 'Target (Integer Value)' input field containing '-1'. Below it is a 'Pre Conditions (For all triggers)' section with a green header labeled 'None'. A red box highlights the '+' button in the top right corner of this section. Below this is a 'Selected Pre Condition Properties' section. On the left, there is a 'Type' dropdown menu with options: 'None' (selected), 'Traveler', and 'Soul'. The 'None' option is highlighted with a teal background.

Target (Integer Value)

-1

Pre Conditions (For all triggers)

+

None

Selected Pre Condition Properties

Type

None

Traveler

Soul

A new condition is created in the pre conditions list when you add a new pre condition. Select the condition in the list, you will be able to define the properties of the pre condition in the Selected Pre Condition Properties section. Select an appropriate condition Type in the dropdown list and configure the specifics. Details of the pre condition properties are explained below:

Pre Condition Type	Description
None	No pre condition is defined.
Traveler	Requires the Traveler XR Attribute to be ON.
Soul	Requires the Soul XR Attribute to be ON.
Debug	Requires the Debug XR Attribute to be ON.
Tracking	Requires the Tracking XR Attribute to be ON.
Stereo	Requires the Stereo XR Attribute to be ON.
Test3D	Requires the Test3D XR Attribute to be ON.
NonTraveler	Requires the Traveler XR Attribute to be OFF.
NonSoul	Requires the Soul XR Attribute to be OFF.
NonDebug	Requires the Debug XR Attribute to be OFF.
NonTracking	Requires the Tracking XR Attribute to be OFF.
NonStereo	Requires the Stereo XR Attribute to be OFF.
NonTest3D	Requires the Test3D XR Attribute to be OFF.
UserLevel	<p>Requires the specified User to be in a certain User level. The options in the UserLevel type are:</p> <ul style="list-style-type: none"> • User Index: index of the User • User Level: required user level (default user level is ranging from 1 to 5) <p>Note: User Level can be set using the vGear.SetUserLevel API.</p>
Context	<p>Requires the VotanicXR application to be in a certain Context mode with Mode Index(int). When no context mode is defined, the default context mode index is 0.</p> <p>Note: Context can be set using the Cmd.Context(int) API.</p>
Tool	<p>Requires the specified controller to have a certain in-game tool active. The options in the Tool type are:</p> <ul style="list-style-type: none"> • Controller Index: defines the index of the virtual controller • Tool Name: defines the name of the tool.
NoObject	Requires the specified controller to be not selecting any interactable objects.
SelectObject	Requires the specified controller to be selecting an interactable object.

TriggerObject	Requires the specified controller to be selecting an interactable object and the “Trigger” command is triggered.
GrabObject	Requires the specified controller to be selecting an interactable object and the “Grab” command is triggered.
Device	Requires the specified virtual controller to be bound to a certain device. The options in the Device type are: <ul style="list-style-type: none"> • Controller Index: defines the index of the virtual controller • Device Name: defines the name of the physical device bind to the virtual controller
Environment	Requires the system configuration’s Environment is set to the specified Environment Type .
CommandReceived	Requires the specified Command Name to be received by this command.

When all the pre-conditions are fulfilled, the command can be invoked if any of the command triggers is triggered.

Selected Command Properties

The screenshot shows the 'Selected Command Properties' interface. At the top, there is a list titled 'Triggers (Command will be sent if ONE is triggered)' with two items: 'Trigger 1' and 'Trigger 2'. Below this is a section titled 'Selected Trigger Properties' containing a 'Conditions' list. The 'Conditions' list has a '+' button highlighted with a red box, and the following items: 'None', 'Traveler', and 'Soul'.

For each trigger, it is possible to define optional trigger conditions that further limits how a command trigger can be triggered. The trigger condition types are identical to those of the pre-condition types, please refer to the pre-condition types table above for details of the condition types.

Command Trigger – Virtual Controller Input Event Mapping

The trigger maps the command to a specified virtual controller input event. Scroll down the command's trigger properties, specifics of the input event mapping can be configured.

Selected Command Properties

Input Controller	0
Input Method	Single
Inputs	<button>+</button> <button>-</button>
ButtonPress: [0]	

Selected Input Properties

Type	ButtonPress
Index	0
Reverse	<input type="checkbox"/> OFF

Details of the trigger – virtual controller input event mapping properties are explained below:

Trigger Property	Description
Input Controller	Defines the index of the virtual controller to listen to the input event. Note: for keyboard events, please put down -1 here.

Input Method	Defines the input event type on the virtual controller. The available input method options are: <ul style="list-style-type: none"> • Single: when a single input event from the specified Inputs is invoked • Double: when the input event from the specified Inputs is invoked twice consecutively • Triple: when the input event from the specified Inputs is invoked three times consecutively • Combo: when all the input events from the specified Inputs are invoked simultaneously
Inputs	Defines a list of input events that can trigger the command trigger. When the Input Method is set to Single / Double / Triple, the command trigger can be triggered if any of the Inputs defined here are invoked. When the Input Method is set to Combo, all Inputs defined should be invoked to trigger the command trigger.
Selected Input Properties	
Type	The type of the input event. The available input event types are: <ul style="list-style-type: none"> • None • ButtonPress: returns true if the button is pressing • ButtonDown: returns true if the button is pressed down • ButtonUp: returns true if the button is released • ButtonHold: returns true if the button is held for a while • ButtonTouch: returns true if the button is touched (for OpenVR controllers) • AxisValue: returns the value of the axis when using the Cmd.Value API • AxisPress: returns true if the axis is pressing • AxisDown: returns true if the axis is pressed down • AxisUp: returns true if the axis is released • AxisHold: returns true if the axis is held for a while • KeyboardPress: returns true if the keyboard key is pressing • KeyboardDown: returns true if the keyboard key is pressed down • KeyboardUp: returns true if the keyboard key is released • KeyboardHold: returns true if the keyboard key is held for a while • MouseAxisValue: returns the value of the mouse axis when using the Cmd.Value API • MousePress: returns true if the mouse button is pressing

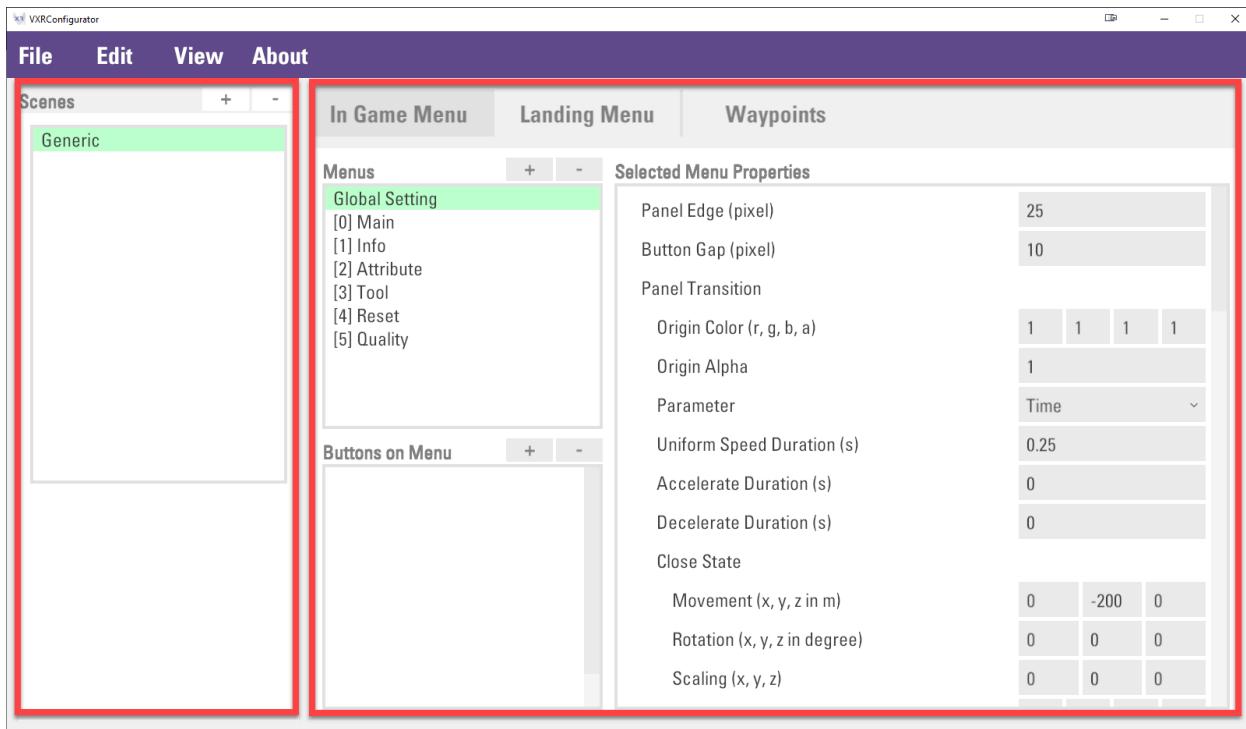
	<ul style="list-style-type: none"> • MouseDown: returns true if the mouse button is pressed down • MouseUp: returns true if the mouse button is released • MouseHold: returns true if the mouse button is held for a while <p>Note: An axis can be used as a button with the AxisPress, AxisDown, AxisUp, and AxisHold input events. The axis is considered as a press when the value of the axis is larger than the Axes Press Detect Value defined for the physical input device (in system configuration's device setting)</p>
Index	Defines the button/axis index of the specified virtual controller.
Key	Defines the name of the key when Keyboard input event types are selected.
Reverse	Toggles to inverse the values received from the input event.
Axis Press Detect Direction	Defines the direction on the axis for listening to the AxisPress , AxisDown , AxisUp , and AxisHold input events.

Application Options Setting

The application options setting file, with file extension “.vxro”, defines additional optional application-specific settings of the VotanicXR application. Properties configured in the application options setting include the in-game menu setting, landing menu setting, and the waypoint setting.

Application Options Setting Overview

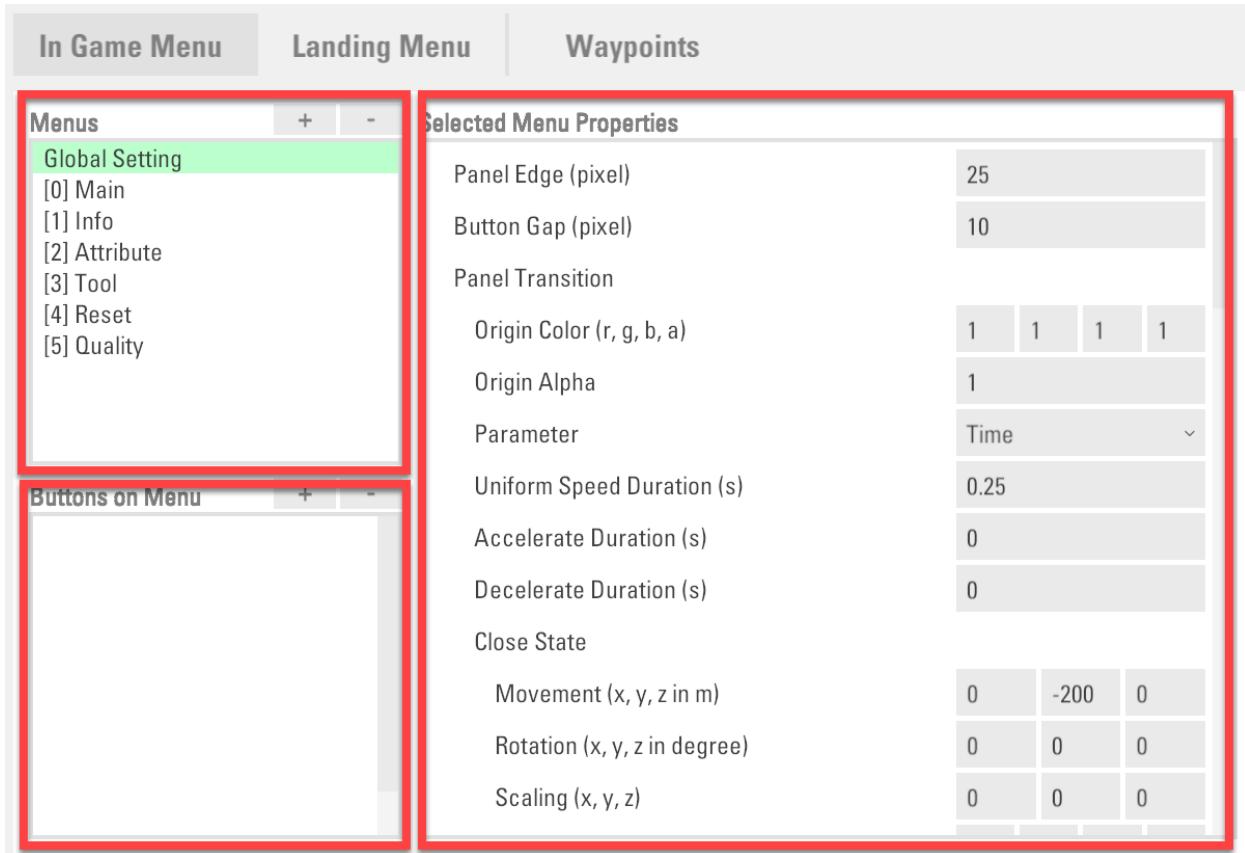
Similar to that of the application settings, the application options setting view in the VotanicXR Configurator is mainly divided into two segments. On the left-hand side is the “Scenes” setting, which specifies the name of the scene in the VotanicXR application in which those settings on the right-hand side apply to. The large area on the right-hand side stores different properties of the application options setting, separated into different groups for ease of accessing different properties.



As the scene profile configuration is already introduced in the application settings section of this guide, the rest of this guide will focus on configuring the application options setting on the right-hand side view.

Configure In-Game Menu

The In Game Menu tab provides an interface to configure VotanicXR's built-in in-game menu for the application. The key concepts, usage, and configuration of the in-game menu is discussed in the “Using In-Game and Landing Menu” tutorial. Below we provide details of the in-game menu properties to supplement the tutorial.



In general, the In Game Menu tab can be divided into three views:

- **Menus View:** defines all the menu panels of the in-game menu.
- **Buttons on Menu View:** defines all buttons of the selected menu panel in the menus view.
- **Selected Menu Properties View:** defines the properties of the selected menu panel or button.

Add / Remove In-Game Menu Item

The steps to add or remove the in-game menu using the VotanicXR configurator is explained in detail in the tutorial, please follow in guides in the tutorial for the process to add or remove a menu item:

- [Add In-Game Menu Item](#)
- [Remove In-Game Menu Item](#)

Configure In-Game Menu Global Setting

The in-game menu global setting is mainly responsible for configuring the look & feel as well as the transition effects of the menu panel and their menu buttons. To configure the global setting of the in-game menu, select the “Global Setting” item at the top of the Menus view and configure the global menu properties on the right.

The screenshot shows the VotanicXR Menus view interface. At the top, there are three tabs: "In Game Menu", "Landing Menu", and "Waypoints". The "In Game Menu" tab is selected. Below the tabs, there are two main sections: "Menus" and "Buttons on Menu". The "Menus" section contains a list of menu items: [0] Main, [1] Info, [2] Attribute, [3] Tool, [4] Reset, and [5] Quality. The item "[0] Main" is highlighted with a red box. The "Buttons on Menu" section is currently empty. On the right side of the interface, there is a large panel titled "Selected Menu Properties" which lists various configuration options with their current values. The properties and values are as follows:

Property	Value
Panel Edge (pixel)	25
Button Gap (pixel)	10
Panel Transition	
Origin Color (r, g, b, a)	1 1 1 1
Origin Alpha	1
Parameter	Time
Uniform Speed Duration (s)	0.25
Accelerate Duration (s)	0
Decelerate Duration (s)	0
Close State	
Movement (x, y, z in m)	0 -200 0
Rotation (x, y, z in degree)	0 0 0
Scaling (x, y, z)	0 0 0

Details of the global menu properties are explained below:

Menu Global Property	Description
Panel Edge	Defines the size of the gap, in terms of pixels, between the outline of the menu button to the edge of the in-game menu object.
Button Gap	Defines the size of the gap, in terms of pixels, between each menu buttons.
Panel Transition Properties	
Origin Color	Defines the original color of the menu panel superimposed on the menu panel's background.
Origin Alpha	Defines the original alpha of the menu panel superimposed on the menu panel's background.
Parameter	<p>Defines the mode of transition when the menu panel transitions from one state to another state. The available transition parameter options are:</p> <ul style="list-style-type: none"> • Time: the time-based transition properties will be used when transitioning. • Velocity: the velocity-based transition properties will be used when transitioning.
When Time is selected	
Uniform Speed Duration	Defines a duration for the menu panel properties to transition from one state to another state uniformly.
Accelerate Duration	Defines a duration for the menu panel properties to transition from one state to another state in increasing speed.
Decelerate Duration	Defines a duration for the menu panel properties to transition from one state to another state in decreasing speed
When Velocity is selected	
Velocity	Defines the maximum rate of transition for the menu panel properties to transition from one state to another state.
Acceleration	Defines the incremental rate on the rate of transition for the menu panel properties to transition from one state to another state.
Deceleration	Defines the decremental rate on the rate of transition for the menu panel properties to transition from one state to another state.
Close State Properties	

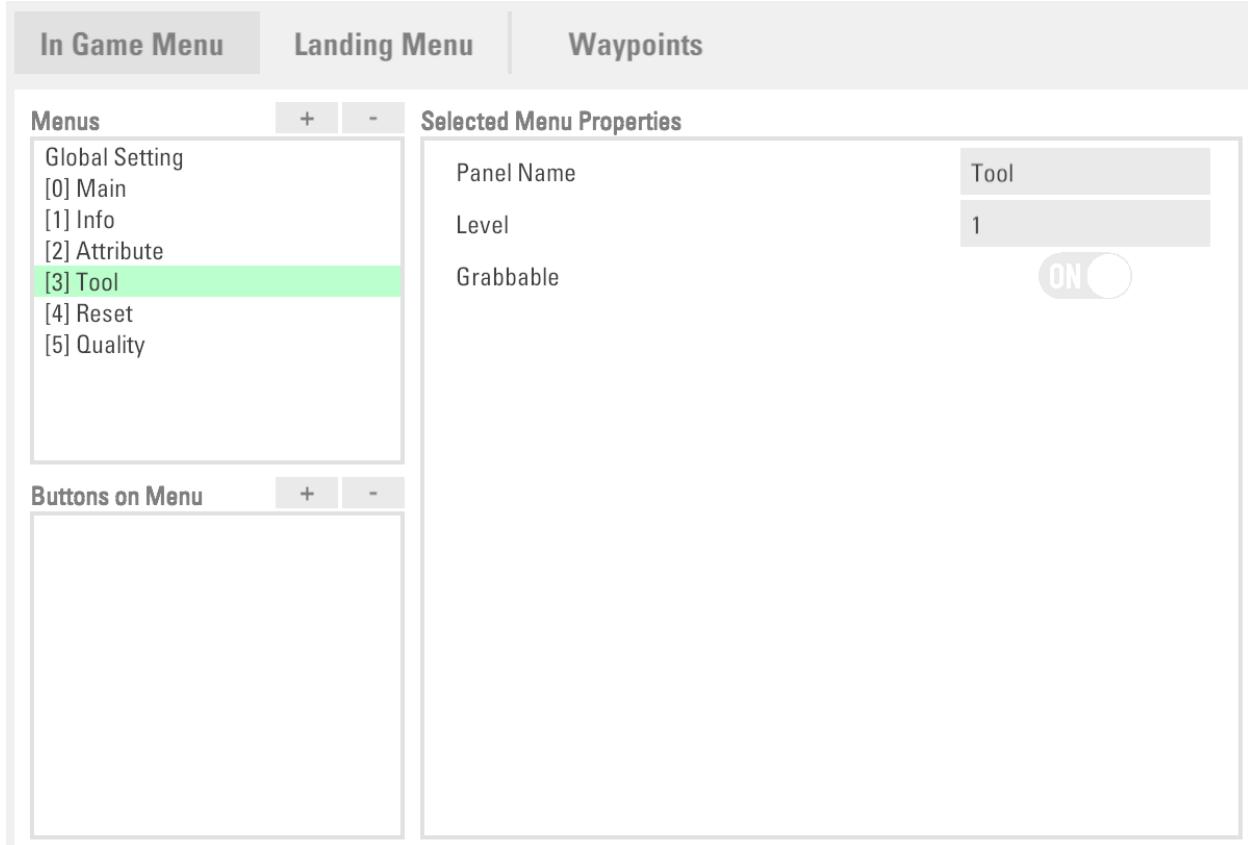
Movement	Defines the movement of the menu panel, in millimeters, when it transitions into the Close state.
Rotation	Defines the rotation of the menu panel, in degrees, when it transitions into the Close state.
Scaling	Defines the change in the menu panel's scale when it transitions into the Close state.
Color	Defines the color of the menu panel when it transitions into the Close state.
Alpha	Defines the alpha of the menu panel when it transitions into the Close state.
Outline	Toggles to enable drawing an outline on the menu panel when it transitions into the Close state.
Color	Defines the color of the outline, if enabled, when it transitions into the Close state.
Width	Defines the width of the outline, if enabled, when it transitions into the Close state.
Disable State Properties	
Movement	Defines the movement of the menu panel, in millimeters, when it transitions into the Disable state.
Rotation	Defines the rotation of the menu panel, in degrees, when it transitions into the Disable state.
Scaling	Defines the change in the menu panel's scale when it transitions into the Disable state.
Color	Defines the color of the menu panel when it transitions into the Disable state.
Alpha	Defines the alpha of the menu panel when it transitions into the Disable state.
Outline	Toggles to enable drawing an outline on the menu panel when it transitions into the Disable state.
Color	Defines the color of the outline, if enabled, when it transitions into the Disable state.
Width	Defines the width of the outline, if enabled, when it transitions into the Disable state.
Button Transition Properties	
Origin Color	Defines the original color of the menu button superimposed on the menu panel.
Origin Alpha	Defines the original alpha of the menu button superimposed on the menu panel.
Parameter	Defines the mode of transition when the menu button transitions from one state to another state. The available transition parameter options are:

	<ul style="list-style-type: none"> • Time: the time-based transition properties will be used when transitioning. • Velocity: the velocity-based transition properties will be used when transitioning. 												
	<p>When Time is selected</p> <table> <tr> <td>Uniform Speed Duration</td><td>Defines a duration for the menu button properties to transition from one state to another state uniformly.</td></tr> <tr> <td>Accelerate Duration</td><td>Defines a duration for the menu button properties to transition from one state to another state in increasing speed.</td></tr> <tr> <td>Decelerate Duration</td><td>Defines a duration for the menu button properties to transition from one state to another state in decreasing speed.</td></tr> </table> <p>When Velocity is selected</p> <table> <tr> <td>Velocity</td><td>Defines the maximum rate of transition for the menu button properties to transition from one state to another state.</td></tr> <tr> <td>Acceleration</td><td>Defines the incremental rate on the rate of transition for the menu button properties to transition from one state to another state.</td></tr> <tr> <td>Deceleration</td><td>Defines the decremental rate on the rate of transition for the menu button properties to transition from one state to another state.</td></tr> </table>	Uniform Speed Duration	Defines a duration for the menu button properties to transition from one state to another state uniformly.	Accelerate Duration	Defines a duration for the menu button properties to transition from one state to another state in increasing speed.	Decelerate Duration	Defines a duration for the menu button properties to transition from one state to another state in decreasing speed.	Velocity	Defines the maximum rate of transition for the menu button properties to transition from one state to another state.	Acceleration	Defines the incremental rate on the rate of transition for the menu button properties to transition from one state to another state.	Deceleration	Defines the decremental rate on the rate of transition for the menu button properties to transition from one state to another state.
Uniform Speed Duration	Defines a duration for the menu button properties to transition from one state to another state uniformly.												
Accelerate Duration	Defines a duration for the menu button properties to transition from one state to another state in increasing speed.												
Decelerate Duration	Defines a duration for the menu button properties to transition from one state to another state in decreasing speed.												
Velocity	Defines the maximum rate of transition for the menu button properties to transition from one state to another state.												
Acceleration	Defines the incremental rate on the rate of transition for the menu button properties to transition from one state to another state.												
Deceleration	Defines the decremental rate on the rate of transition for the menu button properties to transition from one state to another state.												
Select State Properties													
Movement	Defines the movement of the menu button, in millimeters, when it transitions into the Select state.												
Rotation	Defines the rotation of the menu button, in degrees, when it transitions into the Select state.												
Scaling	Defines the change in the menu button's scale when it transitions into the Select state.												
Color	Defines the color of the menu button when it transitions into the Select state.												
Alpha	Defines the alpha of the menu button when it transitions into the Select state.												
Outline	Toggles to enable drawing an outline on the menu button when it transitions into the Select state.												
Color	Defines the color of the outline, if enabled, when it transitions into the Select state.												
Width	Defines the width of the outline, if enabled, when it transitions into the Select state.												

Click State Properties	
Movement	Defines the movement of the menu button, in millimeters, when it transitions into the Click state.
Rotation	Defines the rotation of the menu button, in degrees, when it transitions into the Click state.
Scaling	Defines the change in the menu button's scale when it transitions into the Click state.
Color	Defines the color of the menu button when it transitions into the Click state.
Alpha	Defines the alpha of the menu button when it transitions into the Click state.
Outline	Toggles to enable drawing an outline on the menu button when it transitions into the Click state.
Color	Defines the color of the outline, if enabled, when it transitions into the Click state.
Width	Defines the width of the outline, if enabled, when it transitions into the Click state.
Disable State Properties	
Movement	Defines the movement of the menu button, in millimeters, when it transitions into the Disable state.
Rotation	Defines the rotation of the menu button, in degrees, when it transitions into the Disable state.
Scaling	Defines the change in the menu button's scale when it transitions into the Disable state.
Color	Defines the color of the menu button when it transitions into the Disable state.
Alpha	Defines the alpha of the menu button when it transitions into the Disable state.
Outline	Toggles to enable drawing an outline on the menu button when it transitions into the Disable state.
Color	Defines the color of the outline, if enabled, when it transitions into the Disable state.
Width	Defines the width of the outline, if enabled, when it transitions into the Disable state.

Configure Menu Panel

Properties of a menu panel can be configured by selecting the menu panel in the Menus list and modifying the settings in the Selected Menu Properties view.

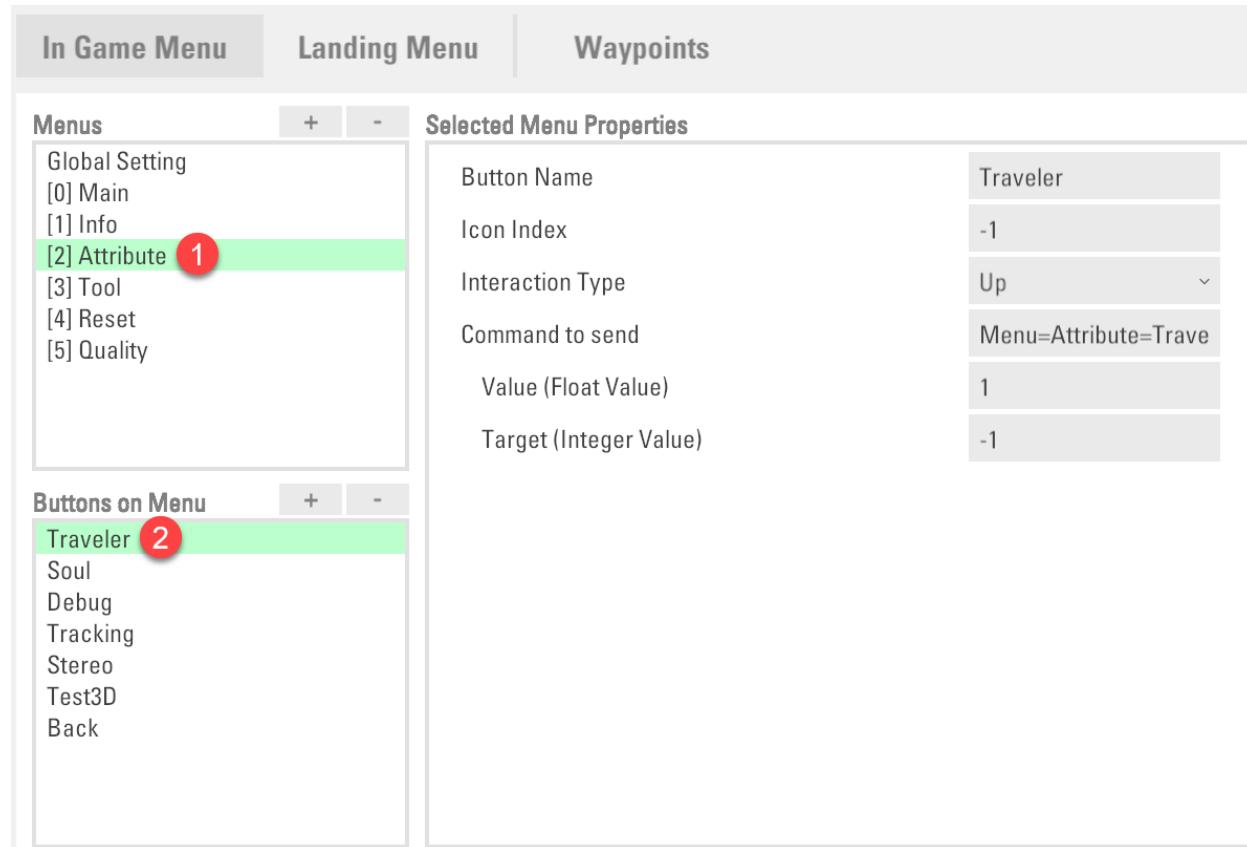


Details of the menu panel properties are explained below:

Menu Panel Property	Description
Panel Name	Defines the name of the menu panel.
Level	Defines the level of the menu panel. A higher-level menu panel will be displayed slightly above the menu panel of a lower level, so that the transition effects between the menu panels will look smoothly.
Grabbable	Toggles to enable the menu panel to be grabbable using the Wand or the Glove.

Configure Menu Button

Properties of a menu button can be configured by first selecting the menu panel in the Menus list then the specific menu button in the Buttons on Menu list, and modifying the settings in the Selected Menu Properties view.



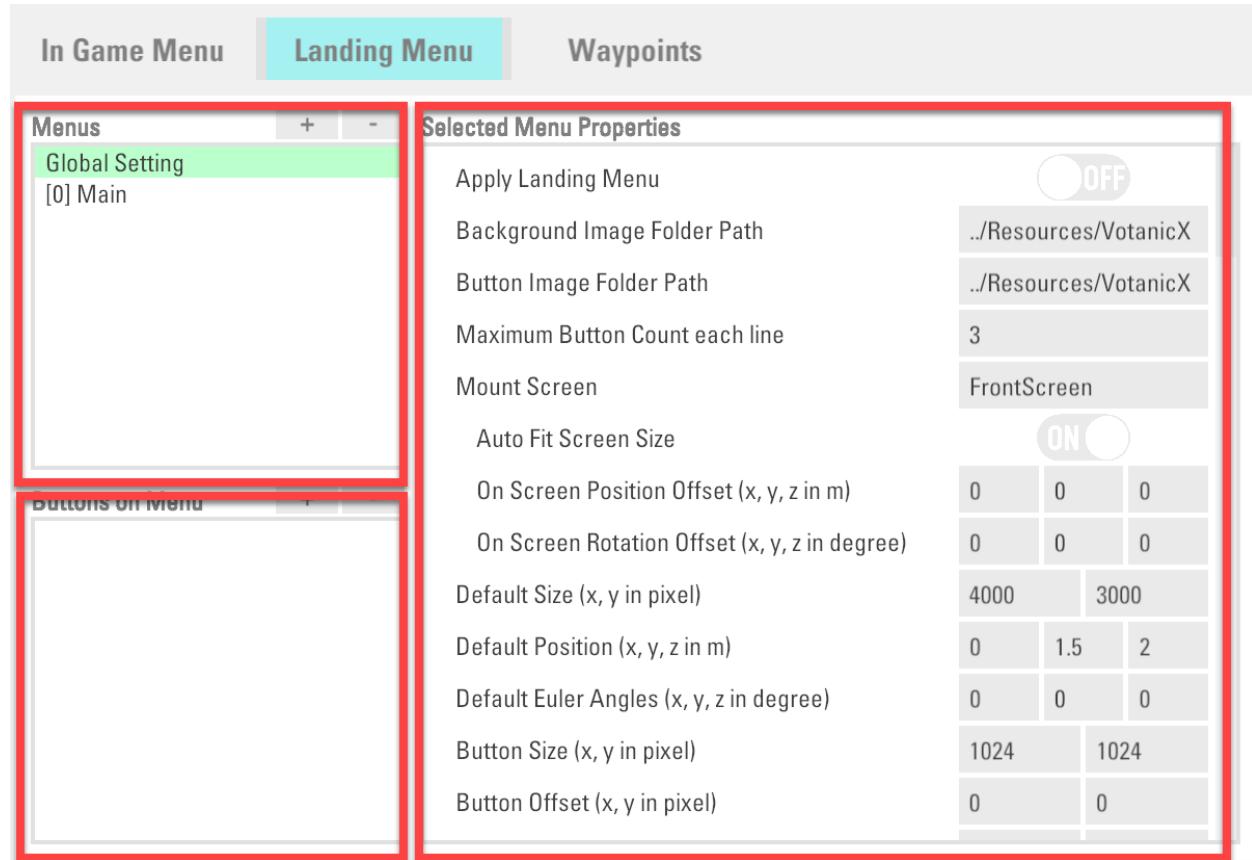
Details of the menu button properties are explained below:

Menu Button Property	Description
Button Name	Defines the name of the menu button. This button name will be shown on the menu panel when the menu panel is displayed in VotanicXR application.
Icon Index	Defines the element index of an icon in the Icons or Core Icons list. Please refer to the tutorial for details of the menu item icon.
Interaction Type	Defines the interaction type for triggering the button command. The default interaction type is “Up”. The Interaction Type is explained in the “Interact with Virtual World” tutorial .

Command to send	Defines the name of the command to send out when the menu button is triggered.
Value	Defines an float value to be sent out when the command is triggered. This value can be received via the when the Cmd.Value API.
Target	Defines an integer value to be sent out with the command when the button is clicked. This value can be received via the API Cmd.Target.

Configure Landing Menu

The Landing Menu is a special type of In-Game Menu, it is also consisting of menu panels with menu buttons on each panel, so the process to configure the landing menu is similar to that of the in-game menu. The basic usage of the landing menu are introduced in the [Landing Menu section of the “Using In-Game Menu and Landing Menu” tutorial](#), this guide supplements the tutorial with details of the landing menu properties.



Similar to the In Game Menu tab, the Landing Menu tab is also divided into the **Menus View**, **Buttons on Menu View**, and **Selected Menu Properties View**. The main different between the landing menu and an ordinary in-game menu lies on its global setting properties, which will be the main focus of this guide.

Configure Landing Menu Global Setting

Other than the basic in-game menu global setting properties, the Landing Menu's global setting adds a few more properties on the physical size and outlook of the menu panel. Details of the landing menu global setting properties are explained below:

Landing Menu Global Property	Description
Apply Landing Menu	Toggles to enable spawning of the landing menu object when the VotanicXR application starts.
Background Image Folder Path	Defines the path of the folder containing all the landing menu background image files. The background images will be loaded and used as background for each landing menu panel, sequentially according to the menu index, in alphabetical order.
Button Image Folder Path	Defines the path of the folder containing all the menu button image files. The button images will be loaded and used as background for each menu button, sequentially according to the button index of each indexed menu panel, in alphabetical order.
Maximum Button Count each line	Defines the maximum number of buttons to display on each row of the menu panel. The button will display on a new row if the total number of buttons exceeds this value.
Mount Screen	Defines the name of the screen (defined in the XR Objects of the system configuration file) in which the landing menu will be displayed on. If the screen is undefined or not available, the default size, position, and Euler angles will be used.
Auto Fit Screen Size	Toggles to enable auto-fitting the landing menu according to the size of the screen defined in the Mount Screen property.
On Screen Position Offset	Defines the position offset of the landing menu with respect to the position of the screen defined in the Mount Screen property.
On Screen Rotation Offset	Defines the rotation offset of the landing menu with respect to the rotation of the screen defined in the Mount Screen property.
Default Size	Defines the default size, in pixels, of the landing menu.
Default Position	Defines the default position of the landing menu.
Default Euler Angles	Defines the default rotation of the landing menu.
Button Size	Defines the size of the buttons on the landing menu.
Button Offset	Defines the position offset of the button on the landing menu.
Button Gap	Defines the size of the gap between each button on the landing menu.
Minimum Button Count for Sliding	Defines the minimum number of buttons on a menu panel to enable the panel slider feature.

Sliding Speed	Defines the speed for the panel slider.
Main Menu Transition Properties	
Origin Color	Defines the original color of the main menu superimposed on the main menu's background.
Origin Alpha	Defines the original alpha of the main menu superimposed on the menu panel's background.
Parameter	<p>Defines the mode of transition when the main menu transitions from one state to another state. The available transition parameter options are:</p> <ul style="list-style-type: none"> • Time: the time-based transition properties will be used when transitioning. • Velocity: the velocity-based transition properties will be used when transitioning.
When Time is selected	Uniform Speed Duration Defines a duration for the main menu properties to transition from one state to another state uniformly.
	Accelerate Duration Defines a duration for the main menu properties to transition from one state to another state in increasing speed.
	Decelerate Duration Defines a duration for the main menu properties to transition from one state to another state in decreasing speed.
	When Velocity is selected
	Velocity Defines the maximum rate of transition for the main menu properties to transition from one state to another state.
	Acceleration Defines the incremental rate on the rate of transition for the main menu properties to transition from one state to another state.
	Deceleration Defines the decremental rate on the rate of transition for the main menu properties to transition from one state to another state.
Close State Properties	
Movement	Defines the movement of the main menu, in millimeters, when it transitions into the Close state.
Rotation	Defines the rotation of the main menu, in degrees, when it transitions into the Close state.
Scaling	Defines the change in the main menu's scale when it transitions into the Close state.

Color	Defines the color of the main menu when it transitions into the Close state.
Alpha	Defines the alpha of the main menu when it transitions into the Close state.
Outline	Toggles to enable drawing an outline on the main menu when it transitions into the Close state.
Color	Defines the color of the outline, if enabled, when it transitions into the Close state.
Width	Defines the width of the outline, if enabled, when it transitions into the Close state.
Disable State Properties	
Movement	Defines the movement of the main menu, in millimeters, when it transitions into the Disable state.
Rotation	Defines the rotation of the main menu, in degrees, when it transitions into the Disable state.
Scaling	Defines the change in the main menu's scale when it transitions into the Disable state.
Color	Defines the color of the menu panel when it transitions into the Disable state.
Alpha	Defines the alpha of the main menu when it transitions into the Disable state.
Outline	Toggles to enable drawing an outline on the main menu when it transitions into the Disable state.
Color	Defines the color of the outline, if enabled, when it transitions into the Disable state.
Width	Defines the width of the outline, if enabled, when it transitions into the Disable state.
Panel Transition Properties	
Origin Color	Defines the original color of the menu panel superimposed on the menu panel's background.
Origin Alpha	Defines the original alpha of the menu panel superimposed on the menu panel's background.
Parameter	<p>Defines the mode of transition when the menu panel transitions from one state to another state. The available transition parameter options are:</p> <ul style="list-style-type: none"> • Time: the time-based transition properties will be used when transitioning. • Velocity: the velocity-based transition properties will be used when transitioning.
	When Time is selected

	Uniform Speed Duration	Defines a duration for the menu panel properties to transition from one state to another state uniformly.
	Accelerate Duration	Defines a duration for the menu panel properties to transition from one state to another state in increasing speed.
	Decelerate Duration	Defines a duration for the menu panel properties to transition from one state to another state in decreasing speed.
	When Velocity is selected	
	Velocity	Defines the maximum rate of transition for the menu panel properties to transition from one state to another state.
	Acceleration	Defines the incremental rate on the rate of transition for the menu panel properties to transition from one state to another state.
	Deceleration	Defines the decremental rate on the rate of transition for the menu panel properties to transition from one state to another state.
	Close State Properties	
	Movement	Defines the movement of the menu panel, in millimeters, when it transitions into the Close state.
	Rotation	Defines the rotation of the menu panel, in degrees, when it transitions into the Close state.
	Scaling	Defines the change in the menu panel's scale when it transitions into the Close state.
	Color	Defines the color of the menu panel when it transitions into the Close state.
	Alpha	Defines the alpha of the menu panel when it transitions into the Close state.
	Outline	Toggles to enable drawing an outline on the menu panel when it transitions into the Close state.
	Color	Defines the color of the outline, if enabled, when it transitions into the Close state.
	Width	Defines the width of the outline, if enabled, when it transitions into the Close state.
	Disable State Properties	
	Movement	Defines the movement of the menu panel, in millimeters, when it transitions into the Disable state.
	Rotation	Defines the rotation of the menu panel, in degrees, when it transitions into the Disable state.

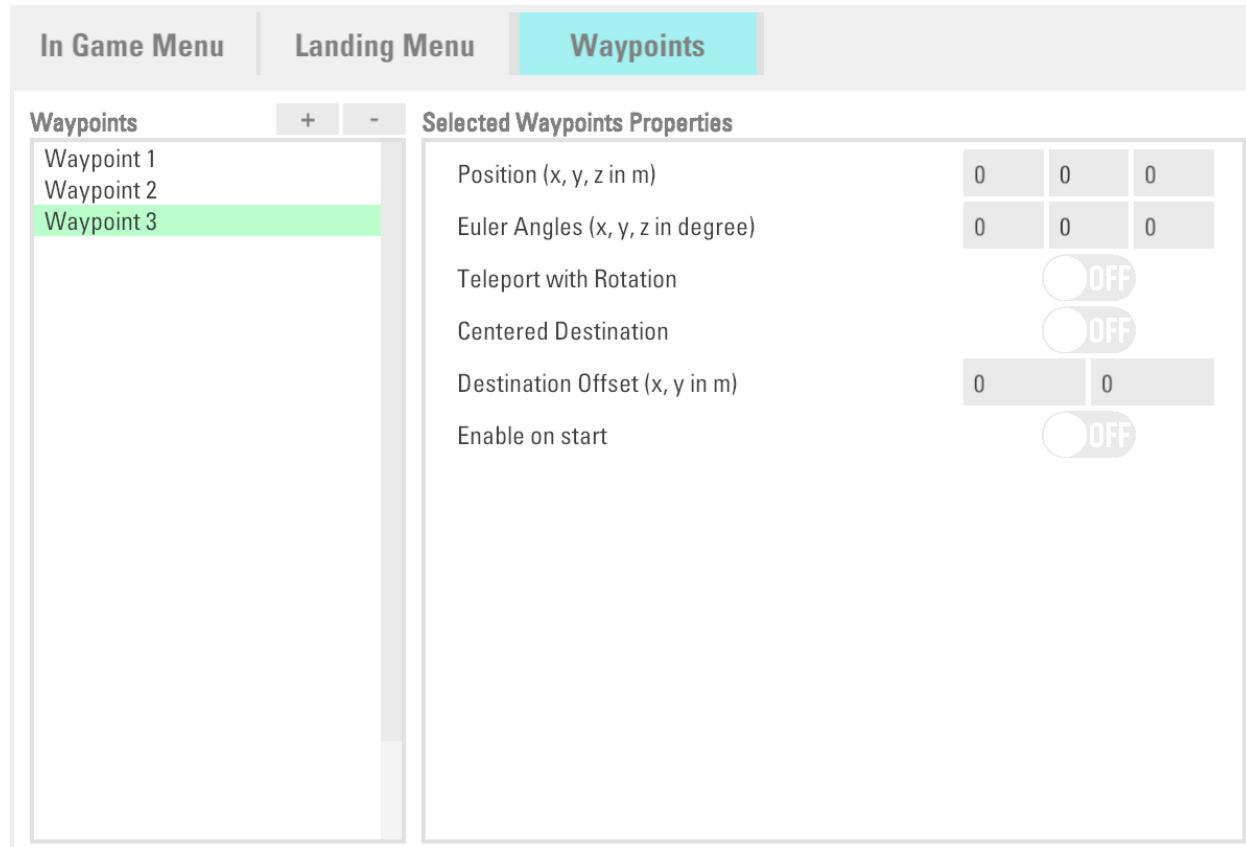
Scaling	Defines the change in the menu panel's scale when it transitions into the Disable state.	
Color	Defines the color of the menu panel when it transitions into the Disable state.	
Alpha	Defines the alpha of the menu panel when it transitions into the Disable state.	
Outline	Toggles to enable drawing an outline on the menu panel when it transitions into the Disable state.	
Color	Defines the color of the outline, if enabled, when it transitions into the Disable state.	
Width	Defines the width of the outline, if enabled, when it transitions into the Disable state.	
Button Transition Properties		
Origin Color	Defines the original color of the menu button superimposed on the menu panel.	
Origin Alpha	Defines the original alpha of the menu button superimposed on the menu panel.	
Parameter	<p>Defines the mode of transition when the menu button transitions from one state to another state. The available transition parameter options are:</p> <ul style="list-style-type: none"> • Time: the time-based transition properties will be used when transitioning. • Velocity: the velocity-based transition properties will be used when transitioning. 	
Parameter = Time Uniform Speed Duration	Defines a duration for the menu button properties to transition from one state to another state uniformly.	
When Time is selected		
Uniform Speed Duration		Defines a duration for the menu button properties to transition from one state to another state in increasing speed.
Accelerate Duration		Defines a duration for the menu button properties to transition from one state to another state in decreasing speed.
Decelerate Duration		Defines the maximum rate of transition for the menu button properties to transition from one state to another state.
When Velocity is selected		

	Velocity	Defines the maximum rate of transition for the menu button properties to transition from one state to another state.
	Acceleration	Defines the incremental rate on the rate of transition for the menu button properties to transition from one state to another state.
	Deceleration	Defines the decremental rate on the rate of transition for the menu button properties to transition from one state to another state.
Select State Properties		
Movement	Defines the movement of the menu button, in millimeters, when it transitions into the Select state.	
Rotation	Defines the rotation of the menu button, in degrees, when it transitions into the Select state.	
Scaling	Defines the change in the menu button's scale when it transitions into the Select state.	
Color	Defines the color of the menu button when it transitions into the Select state.	
Alpha	Defines the alpha of the menu button when it transitions into the Select state.	
Outline	Toggles to enable drawing an outline on the menu button when it transitions into the Select state.	
Color	Defines the color of the outline, if enabled, when it transitions into the Select state.	
Width	Defines the width of the outline, if enabled, when it transitions into the Select state.	
Click State Properties		
Movement	Defines the movement of the menu button, in millimeters, when it transitions into the Click state.	
Rotation	Defines the rotation of the menu button, in degrees, when it transitions into the Click state.	
Scaling	Defines the change in the menu button's scale when it transitions into the Click state.	
Color	Defines the color of the menu button when it transitions into the Click state.	
Alpha	Defines the alpha of the menu button when it transitions into the Click state.	
Outline	Toggles to enable drawing an outline on the menu button when it transitions into the Click state.	
Color	Defines the color of the outline, if enabled, when it transitions into the Click state.	

Width	Defines the width of the outline, if enabled, when it transitions into the Click state.
Disable State Properties	
Movement	Defines the movement of the menu button, in millimeters, when it transitions into the Disable state.
Rotation	Defines the rotation of the menu button, in degrees, when it transitions into the Disable state.
Scaling	Defines the change in the menu button's scale when it transitions into the Disable state.
Color	Defines the color of the menu button when it transitions into the Disable state.
Alpha	Defines the alpha of the menu button when it transitions into the Disable state.
Outline	Toggles to enable drawing an outline on the menu button when it transitions into the Disable state.
Color	Defines the color of the outline, if enabled, when it transitions into the Disable state.
Width	Defines the width of the outline, if enabled, when it transitions into the Disable state.

Configure Waypoints

Waypoints are useful position references used by the teleport tool. The basic usage of waypoints is covered in the [“Locomotion” tutorial](#), which introduced the steps to add and use the Waypoint in Unity Editor. Apart from pre-defining all waypoints when in Unity Editor, waypoints can also be added dynamically by defining their properties in the application options setting file, it is useful when you need to fine-tune the positions of the waypoints without needing to re-build the application. This section introduces the steps to define these waypoints using the VotanicXR configurator.



Waypoints can be added by selecting the “+” button at the Waypoints list and can be configured by selecting the Waypoint name and modifying its properties on the right. Details of the Waypoint property are explained below:

Waypoint Property	Description
Position	Defines the position of the selected waypoint.
Euler Angles	Defines the rotation of the selected waypoint.
Teleport with Rotation	Toggles to enable teleporting to the waypoint using both the waypoint's position and rotation.

Centered Destination	<p>Toggles to enable teleporting the user object transform to the waypoint, so that the waypoint will align with the system's center point after teleport. When it is OFF, the waypoint will align with the user's current standing point in the XR system.</p> <p>Note: The property is useful when standing at the center of the system is the optimum point for viewing the XR content. User shall be instructed to return to standing at the center point of the system after teleporting to the waypoint.</p>
Destination Offset	Defines a position offset from the waypoint in which the user will be teleported to.
Enable on start	Toggles to enable the waypoint to set to “Enable” state at start of the VotanicXR application.