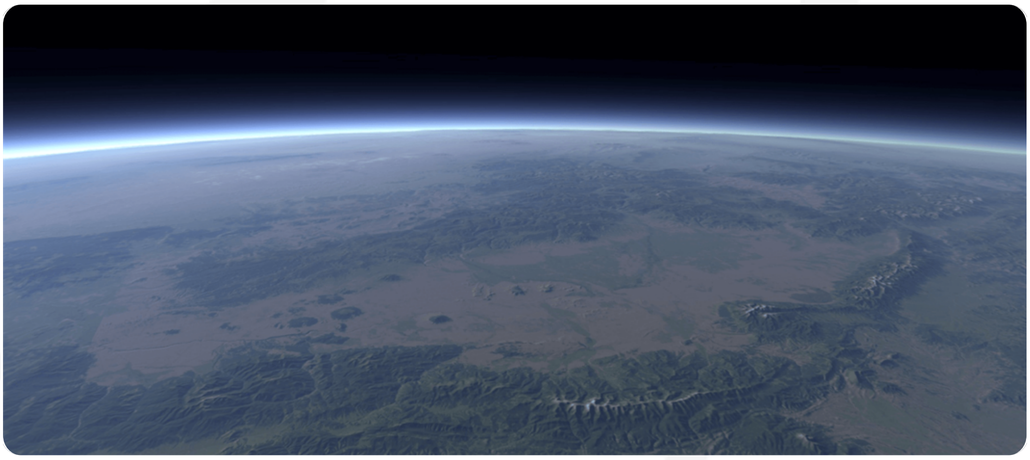
VBS Blue IG



VBS Blue IG 23.2.0

Documentation Legal Notice

Customer Support

The Bohemia Interactive Simulations Support page can be found at:

- http://www.bisimulations.com/support

For any type of assistance with Bohemia Interactive Simulations products, use the following support email and we will respond to your query with urgency.

- support@bisimulations.com

Our website contains a range of media and handouts relating to Bohemia Interactive Simulations products:

- http://www.bisimulations.com/

Contents

1. VBS Blue IG Overview

The VBS Blue Image Generator (VBS Blue IG) provides high-fidelity visualizations of data from VBS4 and Common Image Generator Interface (CIGI) simulations. VBS Blue IG is a whole-earth rendering solution, eliminating the need to limit simulations to the confines of terrain databases. With this software, highly detailed insets can be combined with procedurally generated terrain and vegetation to simulate scenarios anywhere on earth.

WARNING

VBS Blue IG no longer supports VBS3 as a host, and only supports VBS4 as a host, as well as CIGI hosts.

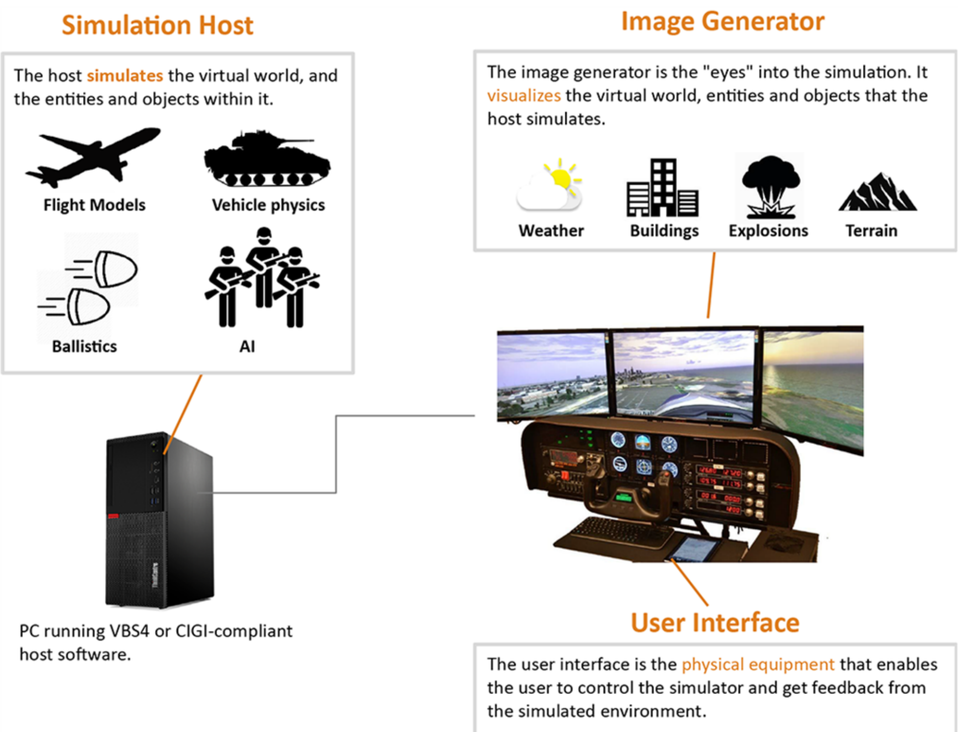
For VBS3 host users, the last supported version is VBS Blue IG 22.1, compatible with VBS3 22.1.

Using the CIGI protocol, a simulation host such as VBS4, or any compliant host connects to and controls the IG.

Designed to support high-performance computer image generation for the full range of military training and special operations applications, VBS Blue IG runs on standard, unmodified COTS hardware or scalable graphics systems.

The figure below illustrates how the simulator and image generator work together.

Image-1: A basic overview of Simulators and Image Generators



VBS Blue IG delivers highly realistic visual and sensor scenes, with long-view distances and large numbers of moving entities on a whole-world terrain, while enabling a single IG to be utilized in Air, Sea and Land domains.

Image-2: Several IG outputs blended to project a seamless image on to a dome

Use the following documentation to guide you through the setup and use of VBS Blue IG.

Deploy VBS Blue IG as described in VBS Blue IG Deployment and Installation in the VBS Blue IG Deployment Guide.

The following documentation helps verify proper deployment of VBS Blue IG:

- CIGI Test Host - Overview (on page 21) - A standalone utility that verifies aspects of a VBS Blue IG deployment, including demos, performance tests, packet communication tests, and content testing.
- Quick Start Guides - Guides to using a simulation host in a basic scenario with VBS Blue IG:
  - Quick Start: CIGI Test Host (on page 14)
  - Quick Start: VBS Blue IG with VBS4 Host (on page 16)
  - Quick Start: Multichannel System (on page 18)

VBS Blue IG uses the following general workflow:

1. Building Projects (on the next page)
2. Use the Runtime Controls (on page 198) for real-time control of settings and cameras in VBS Blue IG clients.

1.1 Building Projects

Review the Setup Guides (on page 22) for step-by-step instructions for the following use cases:

1. Host specific setup:
   - CIGI Test Host - Overview (on page 21)
   - Configure VBS Blue IG and VBS4 (on page 23)
2. Edit settings files directly using Advanced Configuration (on page 434).
3. Complex display solutions:
   - Virtual Reality / Mixed Reality Headsets (on page 62)
   - Warping on Curved Displays (on page 132)

The Interface Control Document contains a detailed explanation of the specific CIGI data packets that VBS Blue IG sends and receives for each supported version of the CIGI specification:

- Interface Control Document - CIGI 4.0 (https://manuals.bisimulations.com/cigiicd/icd_23_2/4.0/overview.html)
- Interface Control Document - CIGI 3.3 (https://manuals.bisimulations.com/cigiicd/icd_23_2/3.3/overview.html)
- Interface Control Document - CIGI 3.2 (https://manuals.bisimulations.com/cigiicd/icd_23_2/3.2/overview.html)
- Interface Control Document - CIGI 3.1 (https://manuals.bisimulations.com/cigiicd/icd_23_2/3.1/overview.html)
- Interface Control Document - CIGI 3.0 (https://manuals.bisimulations.com/cigiicd/icd_23_2/3.0/overview.html)
- Interface Control Document - CIGI 2.1 (https://manuals.bisimulations.com/cigiicd/icd_23_2/2.1/overview.html)
- Interface Control Document - CIGI 2.0 (https://manuals.bisimulations.com/cigiicd/icd_23_2/2.0/overview.html)

> **NOTE**
>
> An offline ICD reference is also available and provided in the following file:
>
> `\IG_Installation\docs\VBS_Blue_IG_ICD_CIGI.zip`
>
> A PDF is also provided in the same folder.

2. Quick Start Guides

This section provides topics on how to quickly configure various systems contained within VBS Blue IG, along with their corresponding configuration and methods of control.

- Quick Start: CIGI Test Host (on the next page) - A summary guide for running the Tests option with demo samples for the CIGI Test Host toolset.
- Quick Start: VBS Blue IG with VBS4 Host (on page 16) - Set up procedure for a basic usage of VBS Blue IG with VBS4.
- Quick Start: Multichannel System (on page 18) - A quick summary for how to set up a multichannel system using VBS Blue IG.

2.1 Quick Start: CIGI Test Host

VBS Blue IG provides a series of demos and test scripts, which can be run from the Tests option of the CIGI Test Host toolset. Use these demo examples to verify that VBS Blue IG is properly installed and functional.

This quick start guide explains how to run an Outpost Assault with four viewports using CIGI Test Host on `localhost`.

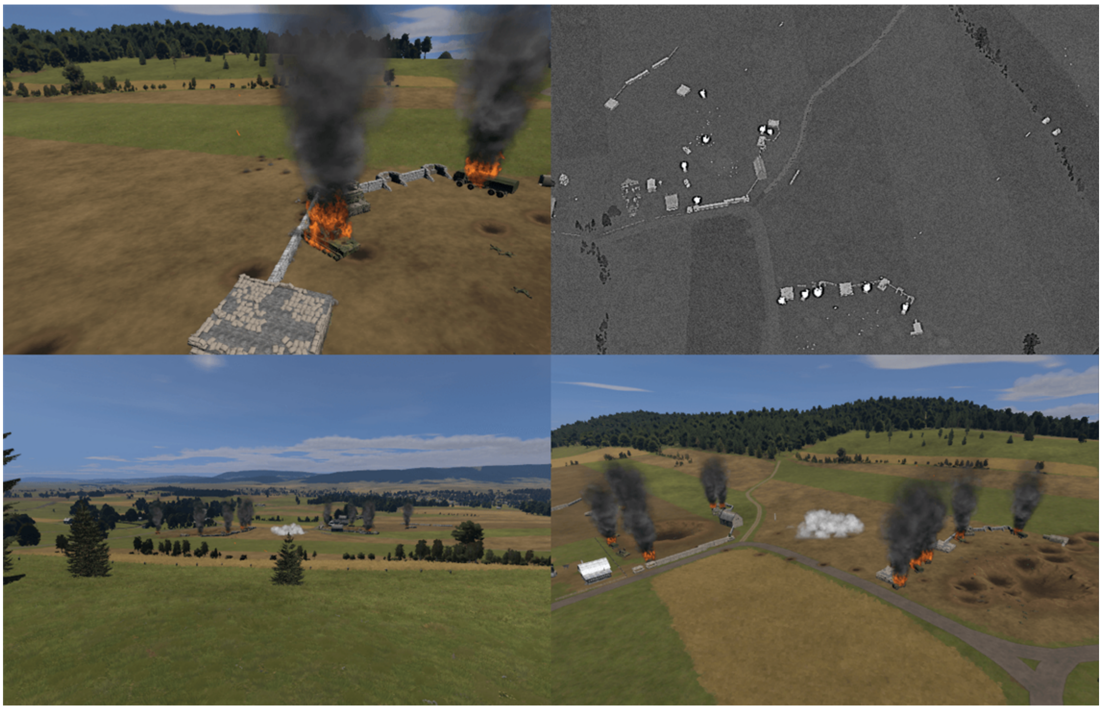
> **NOTE**
> For a more detailed procedure than the summary guide below, including network configuration, see the CIGI Test Host documentation.

Follow these steps:

1. Launch `BlueIG.exe`.
2. Run `CigiTestHost.exe` in:

   `\IG_Installation\tools\CigiTools\`
3. In the CIGI Test Host Home Page, click **Tests**.
4. Expand the **Feature Demos** list.
5. Select **Outpost Assault > Outpost Assault - Four Viewpoints - Day**.
6. In the right pane, click **Run Test**.

   The selected demo runs on the IG client shown in the image below.

7. Select the **Test Log** tab to display information about the test runtime.
8. Click **Pause / Resume Test** to control the test.

> **NOTE**
> The **Resume** button appears once **Pause** is selected.

9. Click **Open CIGI Packet Log Window** to view the packet log.

   For more information, use the CIGI Test Host built-in documentation that can be accessed by the icon in the window title:

10. Click **Stop Test** button to end the demonstration.

Measure the success of a test by successfully receiving and transmitting data packets. In this case, receiving data packets from CIGI Test Host (the simulation host) and transmitting them to VBS Blue IG (the client) with the client using the packets to generate images illustrates a successful setup and testing environment.

2.2 Quick Start: VBS Blue IG with VBS4 Host

This guide explains the most basic usage of VBS Blue IG with VBS4. The following steps cover the creation of a single-view IG project with VBS4 as the host. In this example, both the VBS4 Host and the VBS Blue IG Client run on the same computer.

- VBS4 Computer Setup (below)
- VBS Blue IG Computer Setup (on the next page)

> **NOTE**
> For a more detailed procedure for setting up VBS Blue IG with VBS4, see Configure VBS Blue IG and VBS4 (on page 23).

2.2.1 VBS4 Computer Setup

Start a VBS4 Admin Client to act as the simulation host.

Follow these steps:

1. Use VBS Launcher to start the VBS4 Admin Client with the following parameters selected:
   - In the **VBS4 > Client** tab, select:
     - **VBS4 Offline**
     - **admin**
     - Other Client parameters as required.
   - In the **VBS4 > Server** tab, select **vbsHostNet**.
2. Click **Launch Modules** to start the VBS4 Admin Client.

   For more information, see Starting VBS Blue IG in the VBS4 Administrator Manual.
3. Prepare a Scenario that includes an IG View Object:
   a. Create a Battlespace at a selected location.
   b. Select the Battlespace, highlight **Editor** and click **Create** to open VBS Editor in Prepare mode.
   c. Add a **Unit** and an **IG View Object** to the scenario.
   d. Right-click the IG View Object, select **Link to Unit**, and click the Unit.
   e. Save the Scenario

   For more information, see Scenario Preparation in the VBS4 Editor Manual.

4. **Optional**: Configure the **Multicast TTL (Time To Live)** setting, as needed.
   a. Open the following file:
      `%LOCALAPPDATA%\VBS4\Settings\VBSExternalNetworking.xml`
   b. In the Multicast TTL parameter, adjust the settings, as needed.
      Range of supported values is 0-255, default value is 16.
   c. Save and close the file.
5. Restart VBS4.
6. Execute the Scenario:
   a. Select the Battlespace, highlight **Execute** and click **Host** to open the Network Lobby.
   b. Select the Unit to assign yourself control.
   c. Click **OK**, and then **OK** to start the Scenario.

   For more information, see Scenario Execution in the VBS4 Instructor Manual.

2.2.2 VBS Blue IG Computer Setup

Start VBS Blue IG.

Follow these steps:

1. Launch **BlueIG.exe**.
2. Press **Esc** to show the mouse cursor.
3. Press **Tab** to open the VBS Blue IG **Settings** menu.
4. Select the **VBS External Networking** option, then click **Networking** drop-down arrow.
5. In the **Host Address** text box, input the IP address of the VBS4 computer.
6. Click **Save Settings**.

> **NOTE**
> The Settings folder is only created after a new installation of VBS Blue IG is first launched. Any new settings are applied only after restarting VBS Blue IG.

7. Restart VBS Blue IG.

After VBS Blue IG restarts, it displays the **IG View Object** view, attached to the VBS4 Player unit.

If the view does not display in the expected manner, see Cannot Connect to VBS Host (on page 459).

For a more detailed procedure setting up VBS Blue IG with VBS4 using multiple computers, see Configure VBS Blue IG and VBS4 (on page 23).

2.3 Quick Start: Multichannel System

This quick start guide explains how to set up a multichannel system using VBS Blue IG with CIGI Test Host substituting the simulation host.

The multichannel system includes the following:

- Multiple computers running VBS Blue IG.
  - One instance of VBS Blue IG acts as a Master.
  - Remaining instances serve in the role of Clients.
- A single computer running CIGI Test Host.

The following topics describe the three parts of the setup procedure:

1. Master Instance Setup (below)
2. Client Instance Setup (on the next page)
3. CIGI Test Host Instance Setup (on the next page)

2.3.1 Master Instance Setup

Start a VBS Blue IG client to act as Master.

Follow these steps:

1. Launch `BlueIG.exe`.
2. Press **Esc**, then **Tab** to open the VBS Blue IG Settings (on page 201) menu.
3. Select the **Views** option, and in the right pane, click the **Multichannel** drop-down.
4. In the **General** section, select the **Master** and the **Enabled** checkboxes.

   > **NOTE**
   > Selecting **Master** can be substituted by launching VBS Blue IG with the `-master` startup parameter. For more information, see Startup Parameters (on page 454).

5. Click **Save**.
6. In the VBS Blue IG Settings menu, select the **CIGI** option, then click the **Session #** drop-down menu. For more information, see CIGI Settings (on page 205).
7. In the **Network > Send** section, configure the **Address** field to contain the IP address of the computer running the CIGI Test Host.
8. In the **Network > Receive** section, click the **Multicast Enabled** checkbox.

9. **Optional:** Specify a custom **Multicast Address**.

> **WARNING**
> Use the multicast address specified used in all VBS Blue IG instances in the next steps.

10. Click **Save**.
11. Restart VBS Blue IG.

2.3.2 Client Instance Setup

Start additional VBS Blue IG clients as required.

1. Launch `BlueIG.exe`.
2. Press **Esc**, and then **Tab** to open the VBS Blue IG Settings (on page 201) menu.
3. Select the **Views** option, and in the right pane, click the **Multichannel** drop-down.
4. In the **General** section, click only the **Enabled** checkbox.
5. In the **Networking** section, configure the Master Address field to contain the IP of the computer running the Master VBS Blue IG instance.
6. Click **Save**.
7. With the VBS Blue IG Settings (on page 201) menu still open, select the **CIGI** option.
8. Click the **Session #** drop-down menu.
9. In the **Network > Receive** dropdown, click the **Multicast Enabled** checkbox.
10. **Optional:** Specify a custom **Multicast Address**.

> **NOTE**
> This address must match the address configured on the master instance.

11. In the **Network > Send** section, un-check the **Enabled** checkbox.
12. Click **Save**.
13. Restart VBS Blue IG.

2.3.3 CIGI Test Host Instance Setup

Start CIGI Test Host to act as the host simulation.

1. Run `CigiTestHost.exe` in:

   `\IG_Installation\tools\CigiTools\`

2. Navigate to CIGI Test Host - Settings.
3. In the CIGI Test Host Home page, click **Settings**.
4. Select the **Network** tab to configure the **IP Address** field to match the **Multicast Address** set previously in the VBS Blue IG Master and Client instances.

   > **NOTE**
   > This enables the CIGI Test Host to send information to all VBS Blue IG instances with the same address set in the **Network > Receive** section.

5. Click **Save**.
6. Click **Send packets**.
7. **Create** or **Load** a packet queue, then click **Send Packet Queue** to multicast it to the VBS Blue IG instances.

   > **NOTE**
   > The user can confirm the proper functionality of the system by monitoring the Debug UI. For more information, see IG Multichannel Window (on page 398).

3. CIGI Test Host - Overview

CIGI Test Host is a standalone application for testing CIGI implementation of VBS Blue IG, and is developed by Bohemia Interactive Simulations.

The application provides the following features:

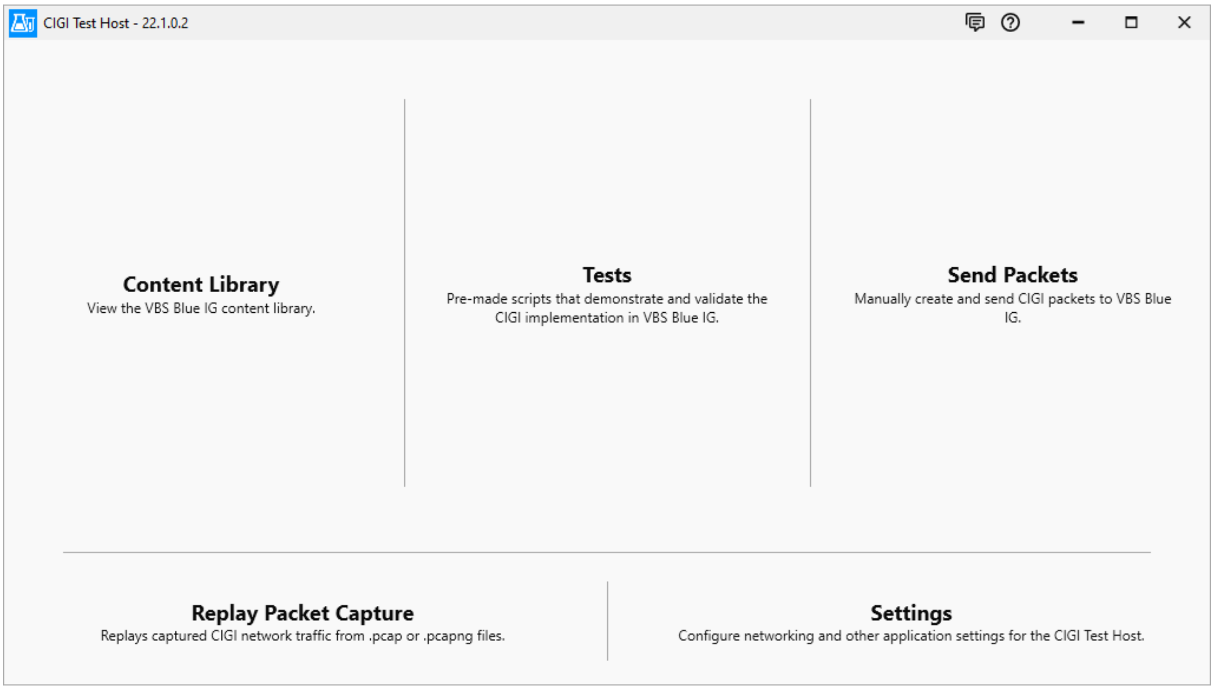
- Send CIGI packets to VBS Blue IG with an easy-to-use graphical interface.
- Create, share, and send a queue of CIGI packets to VBS Blue IG, with additional utilities to create advanced packet queues.
- Run tests and demos to validate and showcase features of VBS Blue IG.
- Review the VBS Blue IG content library, spawn and manipulate models.
- Replay packet capture files (such as those produced by Wireshark) easily, with advanced packet processing and manipulation capabilities.

CIGI Test Host is included with a VBS Blue IG installation and is available in the `\tools\CigiTools\` directory.

Follow these steps:

1. In the *`IG_Installation`*`\tools\CigiTools\` directory, run `CigiTestHost.exe`.

   CIGI Test Host starts and displays its Home page:

2. For further information, use the CIGI Test Host built-in documentation that can be accessed by the icon in the window title:

4. Setup Guides

This section provides specific instructions for using VBS Blue IG with different IG applications (simulation hosts and their related applications) and specialized display equipment.

For setups of VBS Blue IG with IG applications, see:

- Configure VBS Blue IG and VBS4 (on the next page) - VBS4 can serve as the simulation host with VBS Gateway and VBS Host.
- Add IG Viewpoints to Scenarios (on page 35) - Add an IG View Object to a unit or vehicle for each viewpoint that you want to broadcast in a scenario.
- Create IG View Configuration Files (on page 41) - Configure each viewpoint with multiple view perspectives for multiple IG clients.
- Enabling DIS Entities (on page 46) - VBS Set up a scenario to enable DIS entity traffic within a VBS host and VBS Blue IG.
- DIS Bridge (on page 47) - DIS Bridge is a standalone application that handles the merging of simulation information distributed via DIS and CIGI into a single CIGI stream.
- Views Settings (on page 361) - Synchronize multiple channels and designate one of the VBS Blue IG Clients as the master client.

For setups of VBS Blue IG with specialized display equipment, see:

- Virtual Reality / Mixed Reality Headsets (on page 62) - How to use VBS Blue IG with head-mounted displays (HMDs).
- Warping on Curved Displays (on page 132) - Using VBS Blue IG with curved displays, such as domes, based on Generic and Scalable Warping.

4.1 Configure VBS Blue IG and VBS4

This guide explains the process for connecting VBS Blue IG to VBS4. The setup assumes that a licensed and installed version of only one VBS4 Host is running on the network.

> **NOTE**
> If your use case requires running multiple instances of VBS4 and VBS Blue IG, then addresses must be deconflicted. See Address Deconfliction (on page 32) below.

Configuring VBS4 and VBS Blue IG requires multiple steps:

- VBS4 Host Setup (below)
- Preparing Scenarios for VBS Blue IG (on page 25)
- Broadcasting Scenarios to VBS Blue IG (on page 26)
- VBS Blue IG Client Set Up (on page 30)

4.1.1 VBS4 Host Setup

The computer that hosts the Scenario must be configured to broadcast to VBS Blue IG.

There are multiple options for the host:

- Execution using a VBS4 Dedicated Server.
- Execution using a VBS4 Client that also acts as a Host.
- Execution using VBS World Server.

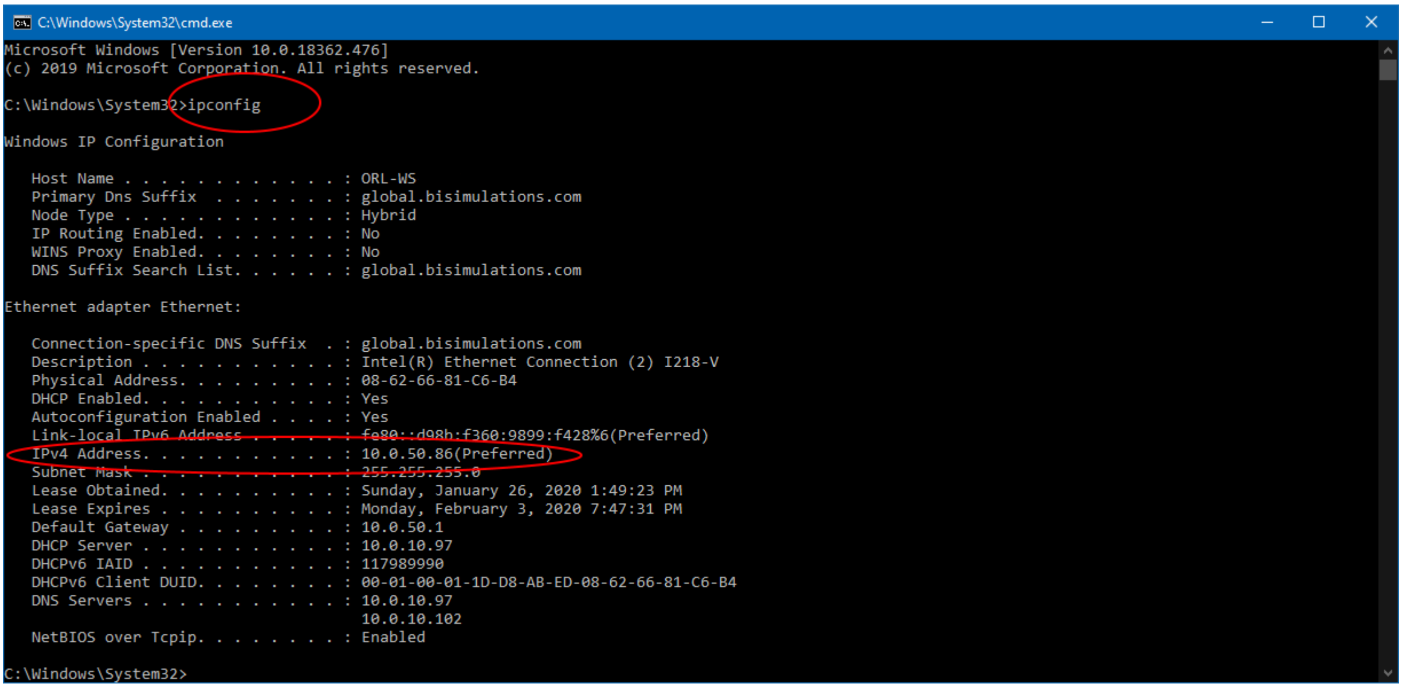
VBS Blue IG setup requires the IP address of the host computer.

Follow these steps:

- In a command prompt window, run the following command to determine the IP address of the host machine:

```
>ipconfig
```

Your window should look similar to the following screen:



Make a note of the **IPv4 Address** which is used later when setting up the connected VBS Blue IG instances.

> **WARNING**
> If the list shows multiple adapters, then setup requires additional steps explained in Address Deconfliction (on page 32).

Modify the VBS4 configuration on the applicable computer.

Follow these steps:

1. **Optional:** Set parameters in the VBS4 configuration file, as required:

   Open the VBS4 configuration file with a text editor:

   `%LOCALAPPDATA%\VBS4\Settings\VBS4.xml`

2. **Optional:** Set up The XRTrainingPlatform (XRTP), a VBS Blue IG component to communicate between VBS4 and VBS Blue IG. The component correlates physical and virtual tracking spaces for Extended Reality (XR). For more information, see XR Training Platform: Overview (on page 86).

VBS4 is configured to act as a host ready to broadcast to VBS Blue IG.

4.1.2 Preparing Scenarios for VBS Blue IG

A VBS4 Scenario for broadcast to VBS Blue IG must contain defined IG Viewpoints.

Follow these steps:

1. Use VBS Launcher to start a VBS4 Admin Client for Scenario Preparation:

   In the **VBS4 > Client** tab, select the VBS4 Configuration to use:

   - **VBS4 Online**

     Starts VBS4 Clients connected to a VBS World Server hosting the Whole-Earth Terrain.

     Click **Refresh**, and select or input the IP Address of a VBS World Server.

   - **VBS4 Offline**

     Starts VBS4 Clients without a connection to a VBS World Server.

     WARNING

     Do not select or input the Server IP Address to connect to a Dedicated Server.

   - Select **admin**.

   NOTE

   Starting VBS4 with `-vbsHostNet` (or legacy `-interopHost`) enabled is not required at this stage.

2. Click **Launch Modules**.

   VBS4 starts in Battlespaces mode.

3. Create a Battlespace that defines the Scenario that you want to broadcast.

   For more information, see Scenario Preparation in the VBS4 Editor Manual.

4. Select the Battlespace, highlight **Editor** and click **Create** to open VBS Editor in Prepare mode.
5. Add a **Unit** to the scenario.
6. Add, configure, and link IG View Objects as described in Add IG Viewpoints to Scenarios (on page 35).
7. Save the Scenario.
8. Close VBS4.

The Scenario is ready to broadcast the defined views when executed on a properly configured host.

For more information, see Scenario Preparation in the VBS4 Editor Manual.

WARNING

If you intend to use a Server to host the Scenario, then copy the Battlespace to the applicable computer before starting the Scenario.

- For Online Execution, use the Synchronize Battlespaces option to copy the Battlespace to the VBS World Server.

  For more information, see Battlespace Management in the Introduction to VBS4 Guide.

- For Offline Execution with a Dedicated Server, copy the Battlespace Folder to the Dedicated Server.

  For more information, see Battlespace Folders in the Introduction to VBS4 Guide.

4.1.3 Broadcasting Scenarios to VBS Blue IG

To broadcast a VBS4 Scenario with IG Viewpoints, start the Host computer configured to broadcast, and use a connected VBS4 Admin Client to start the Scenario.

Only the server should enable networking to VBS Blue IG. All clients will automatically connect as necessary.

Follow these steps:

1. To host the mission, select one of the following options:
   - VBS4 Dedicated Server Setup (Recommended) (on the next page) (recommended)
   - VBS4 Admin Client Host Setup (on page 28)
2. If the Scenario is hosted on an Online VBS World Server or an Offline Dedicated Server, start a VBS4 Admin Client to start the Scenario using one of the following options:
   - Online Use Cases Connected to VBS World Server (on page 28)
   - Offline Use Cases Using Dedicated Server as Host (on page 29)
   - Use Cases with Dedicated Server Running on VBS World Server (on page 30)
3. **Optional parameter**: To allow DIS entity traffic from other simulators besides VBS, set the **interopForwarding** parameter.

   Type the **interopForwarding** parameter in the **Extra parameters** text box.

   For more information, see Enabling DIS Entities (on page 46).
4. **Optional recording / playback of incoming data**: To record and playback incoming traffic on VBS Blue IG from VBS4, follow the steps described in Recording and Playback of VBS External Networking traffic (on page 445).

5. Start VBS4 Trainee clients as required.

   Use the same method described in step 2, but do not select `-admin`.

   To connect to a VBS4 Admin Client running as the host, use the Offline use case and input the IP address of the VBS4 Admin Client as the **Server IP**.

6. On the VBS4 Admin Client, start the Scenario:

   a. Select the Battlespace from the Battlespaces List.

   b. In the Battlespace Functions Panel, under **Execute**, highlight the Scenario name and click **Start**.

      VBS4 opens the Network Lobby and all connected Trainees are also taken to the Network Lobby.

   c. Use the Network Lobby to assign characters, and click **OK**.

   d. In the Mission Briefing, click **OK**.

The Scenario starts and the VBS4 Host broadcasts the defined IG Views.

For more information about running a Scenario, see **Scenario Execution** in the VBS4 Instructor Manual.

4.1.3.1 VBS4 Dedicated Server Setup (Recommended)

1. Open VBS4**Launcher**.

2. In the **VBS4 > Server** tab, select a VBS4**Dedicated Server** from one of the two optional types:

   - **VBS4 Online Dedicated Server** - When selecting this option, click the drop-down menu to choose from an available VBS World Server (VWS).

     For more information, see Connecting Clients in the VBS World Server Manual.

   - **VBS4 Offline Dedicated Server** - When selecting this option, the `-server` parameter appears in the **Parameters** box along with the default settings.

3. Select one of the options for enabling VBS External Networking:
   - Select `-interopHost`, in the Options section of the **VBS4 > Server** tab.
   - Type the parameter `-vbsHostNet` in the **Extra Parameters** text field.

   > **WARNING**
   > Installations of VBS Blue IG versions 20.x and earlier must use the legacy parameter `-interopHost`. For VBS Blue IG versions 21.1 and later, although `-interopHost` is still supported, use the `-vbsHostNet` parameter.

   > **WARNING**
   > When using the VBS4 Dedicated Server option, only this instance of VBS4 should use the parameter `-vbsHostNet`. Any other VBS4 client instances connected to this server should not include the `-vbsHostNet` parameter when setting up, otherwise duplicate entities may appear in the scenario.

4. Select other parameters as required for your use case.
5. Click **Launch Modules**.

   The dedicated server starts in console mode ready to host the scenario and broadcast to VBS Blue IG.

4.1.3.2 VBS4 Admin Client Host Setup

1. Open VBS4**Launcher**.
2. In the **VBS4 > Client** tab, select `-admin`.
3. In the **VBS4 > Server** tab, select `-interopHost`.
4. Select other parameters as required for your use case.
5. Click **Launch Modules**.

   The dedicated server starts in console mode ready to host the scenario and broadcast to VBS Blue IG.

4.1.3.3 Online Use Cases Connected to VBS World Server

For Online use cases connected to VBS World Server use VBS Launcher to start the VBS4 Admin Client with the applicable options:

In the **VBS4 > Client** tab:

- Select **VBS4 Online**.
- Click **Refresh**, and select or input the IP Address of the VBS World Server.

- Select `-admin`.
- Select other parameters as required for your use case.

Click **Launch Modules**.

The VBS4 Client starts ready to start the Scenario.

4.1.3.4 Offline Use Cases Using Dedicated Server as Host

For Offline use cases using a Dedicated Server as the host, use VBS Launcher to start the VBS4 Admin Client with the applicable options:

In the **VBS4 > Client** tab:

- Select **VBS4 Offline**.
- Click **Refresh**, and select or input the **Server IP** address of the Dedicated Server.
- Select `-admin`.
- Select other parameters as required for your use case.

Click **Launch Modules**.

The VBS4 Client starts ready to start the Scenario.

4.1.3.5 Use Cases with Dedicated Server Running on VBS World Server

For use cases with a Dedicated Server running on top of VBS World Server:

- Configure settings for connecting VBS Blue IG to this Dedicated Server (host address and multicast address) in the following XML file:

  `\WS_Installation\Services\VBS4\Profiles\Settings\VBSExternalNetworking.xml`

- To further configure the path to these settings, use the -`profiles=<path>` parameter set in:

  `\WS_Installation\Services\Utilities\_vws_interim_createservices.bat`

4.1.4 VBS Blue IG Client Set Up

The procedure below can be repeated for each VBS Blue IG client connected to the VBS4 Host.

> **NOTE**
> A scenario (or an AAR session) must be running on the VBS4 host before a connection can be successfully established by VBS Blue IG.

Follow these steps:

1. Browse to the VBS Blue IG installation folder.
2. If running with **VBS4 Offline Dedicated Server**, skip to the next step.

   If running with **VBS World Server**, do the following:

   a. Open the file `Run As VWS Client.bat` .

   b. In the line `set VWS_IP=x.x.x.x`, input the IP Address of the VBS World Server.

      Example: `set VWS_IP=192.168.100.200`
3. Launch **BlueIG.exe**.

4. Change the setting for the host IP address using either of the following options:
   a. **Option 1: User Interface configuration**
      i. Click **Tab** on the keyboard to open the **Settings UI**.
      ii. Click **VBS External Networking** in the left pane.
      iii. Click the **Networking** arrow in the right pane to open the networking options.
      iv. In the **Host Address** box, input the IP address of the VBS4 host machine from step 1 of VBS4 Host Setup (on page 23).

      > **NOTE**
      > If you are connecting to VBS World Server, the Host address is the IP of the World Server.

      v. Click **Save Settings** to save the IP address.
      vi. Restart VBS Blue IG to ensure that the new settings take effect.
   b. **Option 2: Edit XML settings file**
      i. Open the file `VBSExternalNetworking.xml` in the following folder:

      `%LOCALAPPDATA%\Bohemia Interactive Simulations\`
      `VBS Blue IG\`*`version`*`\Settings\`

      ii. Find the following line and change the IP address to the address of the VBS4 Host machine:

      `<HostAddress>127.0.0.1</HostAddress>`

      > **NOTE**
      > If you are connecting to a VBS World Server, the Host address is the IP of the World Server.

      iii. Save and close the file.
      iv. Restart VBS Blue IG to ensure that the new settings take effect.

5. Determine the required number of Views or Render Targets for your use case:
    - **Single view setup (default)** - No additional view or render configuration is necessary. The **IG View Object** already added to step 4 in the VBS4 Host Setup (on page 23) section above automatically attaches VBS Blue IG to the view object.
    - **Multiple views or render targets** - For more advanced setups, define the view points for each client:
        - Define the View for each IG Client as described in Views and Render Targets (on page 183).
        - Each IG Client View must correspond to a View defined in VBS4 as described in Create IG View Configuration Files (on page 41).
6. Restart VBS Blue IG to ensure that the new settings take effect.

With the connection established, the IG environment is now controlled by the VBS4 host and the placed **IG View Objects**.

If you have difficulty establishing a connection, see Cannot Connect to VBS Host (on page 459).

4.1.5 Address Deconfliction

Running multiple VBS4 Hosts on the same network requires deconflicting the multicast addresses used by each VBS4 Host.

In order to deconflict the multicast addresses, each additional VBS4 Host should have a unique multicast address. The default address is 225.0.0.1, so additional VBS4 Hosts can simply increment this number.

In the next example, Deconflict VBS4 Host Machine (below), use 225.0.0.2 for the second VBS4 Host.

4.1.5.1 Deconflict VBS4 Host Machine

Follow these steps:

1. Browse to:

   `C:\Users\username\AppData\Local\VBS4\Settings\`.

2. Open the following configuration file:

   `VBSExternalNetworking.xml`

3. Find the following line and change the address to **225.0.0.2**:

```
<MulticastAddress>225.0.0.1</MulticastAddress>
```


4. **Optional step**: Find the following line and change the address to the IP address of the VBS4 host:

   ```
   <MulticastNIC>0.0.0.0</MulticastNIC>
   ```

   > **NOTE**
   > This step is required only if the machines contain multiple network cards (NICs).

   > **NOTE**
   > If you are connecting to a VBS World Server, the NIC address is the IP of the World Server client.

5. Save and close the file.

4.1.5.2 Deconflict VBS Blue IG Machine

Follow these steps:

1. Press **Tab** to open the Settings UI.
2. In the left panel, select VBS External Networking.

   The configuration settings appear in the right panel.
3. In the **Multicast Address** setting box, change the default address (225.0.0.1) to **225.0.0.2.**
4. **Optional step**: In the **Multicast NIC** setting, change the default address (0.0.0.0) to to the IP address of the VBS4 host.

   > **NOTE**
   > This step is required only if the machines contain multiple network cards (NICs).

   > **NOTE**
   > If you are connecting to a VBS World Server, the NIC address is the IP of the World Server client.

5. To save the changes, click **Save Settings**.

   > **WARNING**
   > If you want to revert the changes and restore the original values , click **Revert Changes** before clicking **Save Settings**. Otherwise, if you click **Save Settings** before **Revert Changes**, the changes cannot be reverted.

The VBS Blue IG settings configuration XML files are updated.

6. Repeat steps 1-5 for each VBS Blue IG machine.

4.1.6 Add IG Viewpoints to Scenarios

VBS uses an IG View Object to define a viewpoint in the scenario. The view object is usually linked to a unit or vehicle, and uses a configuration file to determine which IG clients to broadcast to and their individual view perspectives.

Add an IG View Object for each viewpoint that you want to broadcast.

> **NOTE**
> Adding / deleting the IG View Editor Object in VBS Editor during a multiplayer scenario may not be reflected on other clients.

Follow these steps:

1. Open the Scenario to edit in VBS Editor in Prepare mode.
2. Select **IG View Object** from the Editor Objects List.
3. Right-click a location on the map, and select **New Object**.

   The **IG View Object Properties** panel opens.
4. Select the appropriate View **Configuration File**.
5. Click **OK** to add the IG View Object to the map.

Configuration Files	Description
1-Channel Lifeform	A single channel meant to attach to a lifeform, offset at eye-level.
3-Channel AH-64 - Attach Example	3 channels, ID 0-2, offset to display the cockpit view of the AH-64.
3-Channel Lifeform	3 channels, ID 0-2, meant to attach to a lifeform, offset at eye-level.
3-Channel M1A1 - Attach Example	3 channels, ID 0-2, with views attached to various memory points on an M1A1.
3-Channel M1A1 - Main Turret	3 channels, ID 0-2, configured in a widescreen format; attached to the tip of the main turret of an M1A1.
3-Channel Third Person - Far	3 channels, ID 0-2, configured in a widescreen format; meant to attach to a lifeform, offset 50 meters behind the unit.
3-Channel Third Person - Near	3 channels, ID 0-2, configured in a widescreen format; meant to attach to a lifeform, offset 15 meters behind the unit.
3-Channel Third Person - Top Down	3 channels, ID 0-2, configured in a widescreen format; meant to attach to a lifeform, offset 50 meters above the unit looking down.
4-Channel Lifeform (FOV)	4 channels, ID 0-3, configured in a square format; meant to demonstrate advanced frustum control.

Configuration Files	Description
4-Channel Lifeform (YPR)	4 channels, ID 0-3, configured in a square format; meant to demonstrate rotation control.
14-Channel Scalable	14 channels, ID 0-13, used to demonstrate view groupings and scalable configuration.
Custom	Add your own View Configuration files as described in Create IG View Configuration Files (on page 41). **WARNING** The required view configuration file must exist in the following folder: `\VBS_Installation\Settings\CIGI\Views\`

Link the IG View Object to the entity that represents its viewpoint:

1. Right-click the **IG View Object.**
2. Select the appropriate **Link To** option.
3. Click the required entity.

When the scenario runs, the viewpoint defined by the IG View Object moves with the linked entity, and VBS broadcasts the defined views to the specified IG clients.

4.1.6.1 Multiple IG View Objects

Use of multiple IG View Objects in a scenario requires the creation of additional IG view configuration files. In many cases, existing configuration files can be copied, renamed, and edited.

The following example covers a scenario in which 3 IG view objects are attached to entities in a scenario, with the `1-Channel Lifeform.xml` configuration file as the base for the configuration file for each view object.

Follow these steps:

1. On the host computer, open the Views directory located in:

   `\VBS_Installation\Settings\CIGI\Views\`

2. Select `1-Channel Lifeform.xml` and right-click on it.

   Copy it and paste it twice (for a total of 3 instances of the `1-Channel Lifeform.xml` file).

3. Rename the configuration files to:

   - `1-Channel Lifeform-0.xml`
   - `1-Channel Lifeform-1.xml`
   - `1-Channel Lifeform-2.xml`

4. Open `1-Channel Lifeform-0.xml` with a text editor.
5. Go to line 4 (`<ViewGroup>` parameter) and change `<ID>1</ID>` to `<ID>100</ID>`.

> **NOTE**
> This value is arbitrary, but must be unique for each view in the IG project.

6. Go to line 15 (`<View>` parameter) and change `<ID>1</ID>` to `<ID>0</ID>`.
7. Go to line 16 (`<View>` parameter) and change `<GroupID>1</GroupID>` to `<GroupID>100</GroupID>`.

> **NOTE**
> This value must match the value set on line 4.

```
<?xml version="1.0"?>
<View_Config>
  <ViewGroup>
    <ID>100</ID> <!-- Matches the <View> GroupID -->
    <AttachTo>0</AttachTo>
    <Yaw_Offset>0</Yaw_Offset>
    <Pitch_Offset>0</Pitch_Offset>
    <Roll_Offset>0</Roll_Offset>
    <X_Offset>0</X_Offset>
    <Y_Offset>0</Y_Offset>
    <Z_Offset>0</Z_Offset>
    <Precipitation_Radius>1</Precipitation_Radius>
  </ViewGroup>
<View>
  <ID>0</ID> <!-- Matches filename number (0 in this case) -->
  <GroupID>100</GroupID> <!-- Matches the <ViewGroup> ID -->
  <AttachTo>0</AttachTo>
```

8. Save and close the file.
9. Open `1-Channel Lifeform-1.xml` with a text editor.
10. Go to line 4 (`<ViewGroup>` parameter) and change `<ID>1</ID>` to `<ID>200</ID>`.

> **NOTE**
> This value is arbitrary, but must be unique for each view in the IG project.

11. Go to line 16 and change `<GroupID>1</GroupID>` to `<GroupID>200</GroupID>`.

    > **NOTE**
    > This value must match the value set on line 4 (`<ViewGroup>` parameter).

    > **NOTE**
    > Because the default value for `<ID>` on line 15 is 1, no change needs to be made to that setting in `1-Channel Lifeform-1.xml`.

12. Save and close the file.
13. Open `1-Channel Lifeform-2.xml` with a text editor.
14. Go to line 4 and change `<ID>1</ID>` to `<ID>300</ID>`.

    > **NOTE**
    > This value is arbitrary, but must be unique for each view in the IG project.

15. Go to line 15 (`<View>` parameter) and change `<ID>1</ID>` to `<ID>2</ID>`.
16. Go to line 16 and change `<GroupID>1</GroupID>` to `<GroupID>300</GroupID>`.

    > **NOTE**
    > This value must match the value set on line 4.

17. Save and close the file.
18. Launch VBS Blue IG and the VBS Blue IG clients.
19. Enter a mission.
20. Open VBS Editor in Execute (RTE) mode.
21. Place an IG View Object in the scenario.

    The **Object Properties** panel appears.
22. Select `1-Channel Lifeform-0.xml` from the **Configuration Files** drop-down and click **OK**.
23. Place an entity in the scenario.
24. Link the IG View Object to the entity.
25. Create and link 2 more IG View Objects and entities by repeating steps 21-24, but select `1-Channel Lifeform-1.xml` and then `1-Channel Lifeform-2.xml` when placing the IG View Objects.

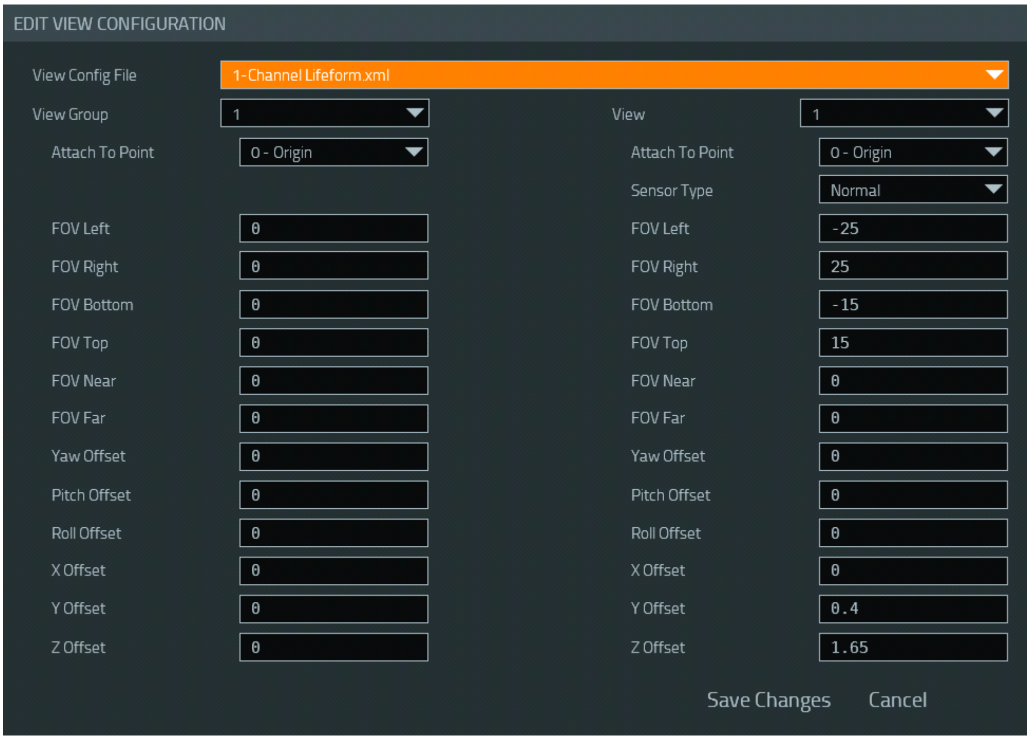
26. For each VBS Blue IG instance, set the View Configuration file to include the View IDs (<View><ID>) established above:
    a. Open the xml file located in the *\IG_Installation* folder:
       \components\BlueIGViewSystems\Config\DefaultViewConfig.xml
    b. Set the required <View><ID> parameter to match all View IDs created for the VBS4 host.
    c. Set any **Optional Fields**, as needed.

> **NOTE**
> For a more detailed procedure for setting up the VBS Blue IG View configuration file, see Views and Render Targets (on page 183).

4.1.6.2 Edit IG View Configurations

The IG View Editor allows users to modify the settings of IG View Configuration files in real-time, and is accessible in Execute (RTE) mode and C2 modes. The parameters contained in IG View configurations are explained in the Create IG View Configuration Files (on page 41) topic.

Access the **IG View Editor** from the VBS Editor **Tools** menu.



Follow these steps:

1. Select the desired configuration file from the **View Config File** drop-down menu.
2. Enter the desired values into any fields you need to modify.

For more information, see View Parameters (on page 42).

3. Click **Save Changes**.

If the edited IG View Configuration file is currently in use by an IG View object, the view linked to that IG View object refreshes to display the view with updated parameters.

4.1.6.3 Sensors

IG views can be modified to display as various types of sensors. Explore the following script commands to modify sensor views:

- `IG_ViewSetSensor` - Use this SQF command to change the sensor type for each defined view individually.

  A detailed explanation and example usage can be found in the VBS Scripting Reference:

  IG_ViewSetSensor (https://sqf.bisimulations.com/display/SQF/IG_ViewSetSensor)

- `IG_ViewSetSensorParameters` - Use this SQF command to further modify parameters of each sensor.

  A detailed explanation and example usage can be found in the VBS Scripting Reference:

  IG_ViewSetSensorParameters (https://sqf.bisimulations.com/display/SQF/IG_ViewSetSensorParameters)

4.1.7 Create IG View Configuration Files

VBS uses XML files to configure each viewpoint with multiple view perspectives for multiple IG clients. Each configuration defines a single viewpoint in VBS, that may contain multiple view perspectives.

View configuration files must be placed in the following folder:

`\VBS_Installation\Settings\CIGI\Views\`

Within the file, the viewpoint is defined by a `<View_Config>` tag which contains `<View>` tags for each perspective. The property tags within view define the orientation, angle, position, and size of the view as well as the IG client to broadcast to.

Follow these steps:

1. Open a new `.xml` file in a text editor.
2. Add `<View_Config>` and `</View_Config>` tags as the first and last lines of the file respectively.
3. Add a `<View>` and `</View>` tag for each perspective that the configuration needs to broadcast to an IG client.
4. Within each `<View>` tag, specify View Parameters (on the next page) to define its perspective.
5. Save the file to the `\Views\` folder:

   `\VBS_Installation\Settings\CIGI\Views\`
6. Copy this file to the same `\Views\` folder for all VBS instances in the network running in administrator mode.

   This configuration file must also be saved into the `\Views\` folder for any host device acting as the **vbsHostNet** for the IG client. The **vbsHostNet** device will likely be the **Dedicated Server** or **VWS** on a VBS network deployment, where applicable.

   > **WARNING**
   > Each defined view must correspond to a configured View on an IG Client.
   >
   > To define view configurations on VBS Blue IG Clients, see Views and Render Targets (on page 183).

   For more information about configuring the IG Client, see Quick Start: VBS Blue IG with VBS4 Host (on page 16).

The View Configuration file is ready to use to define an IG View Object as described in Add IG Viewpoints to Scenarios (on page 35).

4.1.7.1 View Parameters

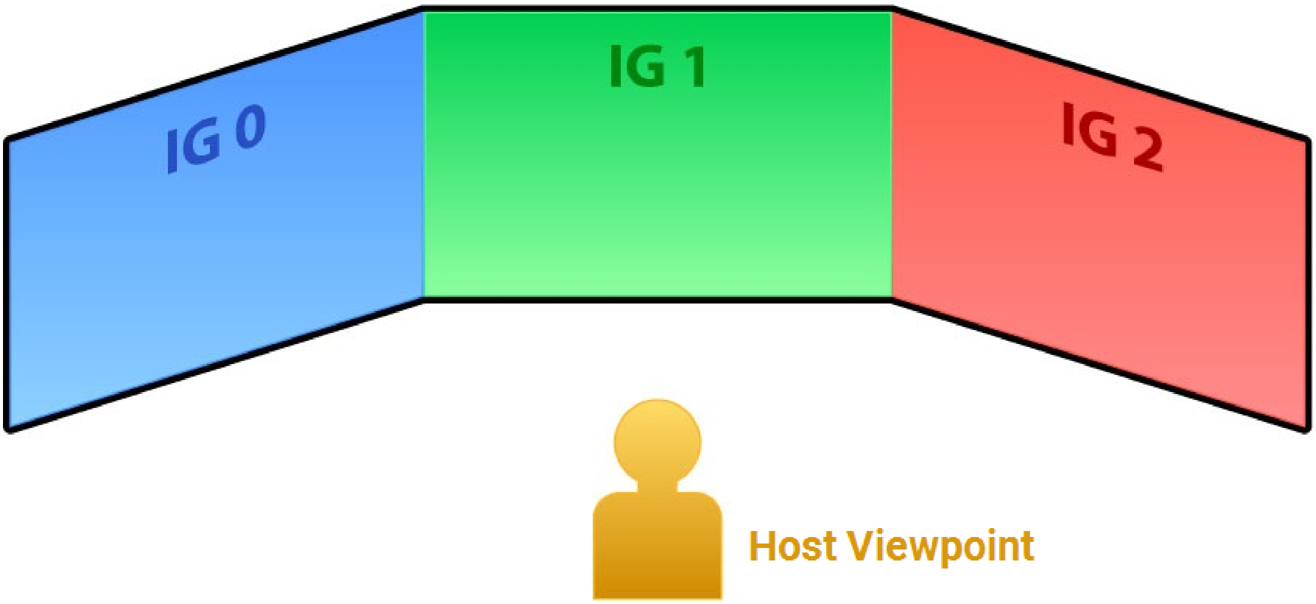
Modify the following parameters of a Viewpoint `.xml` file to define each View perspective:

XML Element	Description
`<ID>`	The unique ID for the IG client, between 0 and 65535.
`<AttachTo>`	This determines which part of the object / vehicle to use to attach the view: 0 - hull 1 - main turret (azimuth) 2 - main turret (elevation) 3 - commander turret (azimuth) 4 - commander turret (elevation) 5 - loader turret (azimuth) 6 - loader turret (elevation) 7 - first-person view (Use when attaching to a unit.) Not all vehicles have these parts. Attaching to an invalid part locks the camera to the entity origin with the camera orientation facing north.
`<AttachToBone>`	Specifies a bone by name to attach the view to. This field is ignored if empty or not present.
`<GroundClampType>`	0 - Default - Default ground clamping applied as currently configured in the IG settings. 1 - None - Ground clamping disabled. 2 - Clamp - Height ground clamping only. 3 - Conform - Conformal ground clamping only. 4 - ClampAndConform - Height and conformal ground clamping.
`<SmoothType>`	0 - Default - Default smoothing applied as currently configured in the IG settings. 1 - Disabled - Smoothing disabled. 2 - Enabled - Smoothing enabled. **NOTE** These parameters are used to override ground clamping and smoothing values set in VBS Blue IG Settings. They only take effect when the IG View Object is attached to a vehicle (so no effect if the view object is not attached to anything or is attached to a lifeform).

XML Element	Description
`<ParentDrawMode>`	Specifies the displayed draw mode of the vehicle the IG View Object is attached to. This has no effect if the view obj is unattached. (For example, if the view is attached to a tank and the **Pilot** draw mode is provided, then the tank will be rendered in the Pilot draw mode for this view). 0 - Normal - The default view geometry that a vehicle is created with. 1 - Pilot - The visual geometry that can be seen from the pilot / driver position of a vehicle. 2 - Gunner - The visual geometry that can be seen from the gunner position of a vehicle. 3 - Cargo - The visual geometry that can be seen from the cargo position of a vehicle.
`<CrewPosition>`	Specifies a crew position by ID to attach the view to. This has no effect if the view obj is unattached. 0 - Pilot / driver, the remaining indices are vehicle-dependent. -1 - None
`<FOV_Left>`	Left half-angle of the view frustum in degrees.
`<FOV_Right>`	Right half-angle of the view frustum in degrees.
`<FOV_Bottom>`	Bottom half-angle of the view frustum in degrees.
`<FOV_Top>`	Top half-angle of the view frustum in degrees.
`<FOV_Near>`	Near clipping plane of the view frustum.
`<FOV_Far>`	Far clipping plane of the view frustum.
`<Yaw_Offset>`	Angle of clockwise rotation around the Up vector in degrees.
`<Pitch_Offset>`	Angle of clockwise rotation around the Left vector in degrees.
`<Roll_Offset>`	Angle of clockwise rotation around the Forward vector in degrees.
`<X_Offset>`	Distance from the entity origin along the Right vector in meters.
`<Y_Offset>`	Distance from the entity origin along the Forward vector in meters.
`<Z_Offset>`	Distance from entity origin along the Up vector in meters.

4.1.7.2 Viewpoint Example

A viewpoint with three perspectives, each displaying 50 degrees of a 150 degree field of view:



`<View_Config>` **and** `<View>` **definitions:**

```
<View_Config>
  <View>
    <ID>0</ID>
    <AttachTo>0</AttachTo>
    <FOV_Left>-25</FOV_Left><!-- negative, based on forward orientation -->
    <FOV_Right>25</FOV_Right><!-- positive, based on forward orientation -->
    <FOV_Bottom>-15</FOV_Bottom><!-- negative, based on forward orientation -->
    <FOV_Top>15</FOV_Top><!-- positive, based on forward orientation -->
    <FOV_Near>0</FOV_Near>
    <FOV_Far>0</FOV_Far>
    <Yaw_Offset>-50</Yaw_Offset>
    <Pitch_Offset>0</Pitch_Offset>
    <Roll_Offset>0</Roll_Offset>
    <X_Offset>-0.85</X_Offset>
    <Y_Offset>-0.1</Y_Offset>
    <Z_Offset>1.55</Z_Offset>
  </View>
  <View>
    <ID>1</ID>
    <AttachTo>0</AttachTo>
    <FOV_Left>-25</FOV_Left><!-- negative, based on forward orientation -->
    <FOV_Right>25</FOV_Right><!-- positive, based on forward orientation -->
    <FOV_Bottom>-15</FOV_Bottom><!-- negative, based on forward orientation -->
```

```
    <FOV_Top>15</FOV_Top><!-- positive, based on forward orientation -->
    <FOV_Near>0</FOV_Near>
    <FOV_Far>0</FOV_Far>
    <Yaw_Offset>0</Yaw_Offset>
    <Pitch_Offset>0</Pitch_Offset>
    <Roll_Offset>0</Roll_Offset>
    <X_Offset>-0.85</X_Offset>
    <Y_Offset>-0.1</Y_Offset>
    <Z_Offset>1.55</Z_Offset>
  </View>
  <View>
    <ID>2</ID>
    <AttachTo>0</AttachTo>
    <FOV_Left>-25</FOV_Left><!-- negative, based on forward orientation -->
    <FOV_Right>25</FOV_Right><!-- positive, based on forward orientation -->
    <FOV_Bottom>-15</FOV_Bottom><!-- negative, based on forward orientation -->
    <FOV_Top>15</FOV_Top><!-- positive, based on forward orientation -->
    <FOV_Near>0</FOV_Near>
    <FOV_Far>0</FOV_Far>
    <Yaw_Offset>50</Yaw_Offset>
    <Pitch_Offset>0</Pitch_Offset>
    <Roll_Offset>0</Roll_Offset>
    <X_Offset>-0.85</X_Offset>
    <Y_Offset>-0.1</Y_Offset>
    <Z_Offset>1.55</Z_Offset>
  </View>
</View_Config>
```

4.1.8 Enabling DIS Entities

VBS scenarios broadcasting to VBS Blue IG may also be communicating with other DIS-compliant simulation products hosting external entities.

Enabling DIS entity traffic within a VBS host and VBS Blue IG configuration requires the **interopForwarding** option to be enabled.

Follow these steps:

1. With a text editor, open the **VBS4.xml** file in the AppData Local folder at:
   - Default VBS4 Profile location:

     `%LOCALAPPDATA%\VBS4\Settings\VBS4.xml`
   - Other VBS4 Profile location:

     `Path\Settings\VBS4.xml`

     Where `Path` is specified using the `-profiles=Path` command-line option.

     For more information, see Command Line and Launcher Options in the VBS Blue IG Administrator Manual.
2. Modify the existing **interopForwarding** entry or add if not present.

   Use the following snippet to the `<Uncategorized>` section of the XML file:

```
<Value>
    <Name>interopForwarding</Name>
    <Value>1</Value>
</Value>
```

3. Save and close the file.

4.2 DIS Bridge

DIS Bridge is a standalone application that handles the merging of simulation information distributed via DIS and CIGI into a single CIGI stream. Other features of the DIS-CIGI Bridge include:

- Custom assignment of CIGI entities to DIS detonations.
- Custom mappings between DIS & CIGI entity types.
- Specification of weapon and articulated part information for given DIS entity types

The application can be used in two modes depending on user needs:

- **DIS only mode** - This method converts a DIS stream into CIGI data.
- **CIGI Bridge mode** - This method acts similarly to **DIS only mode**, but combines an incoming CIGI Stream with the converted DIS > CIGI traffic.

The behavior of the application can be configured through XML files found in the `\IG_Installation\DISCIGIBridge\binaries\Settings` folder, or by using the graphical user interface. All settings are available for configuration through either method.

Follow the procedures below to set up DIS Bridge.

- How to use different bridge modes (below)
- Configuration (on page 49)
- Reference (on page 51)

4.2.1 How to use different bridge modes

- DIS > CIGI mode (below)
- Bridge mode (on the next page)

4.2.1.1 DIS > CIGI mode

Use this mode to convert a DIS stream into CIGI data.

Follow these steps:

1. Start the application directly by opening the .exe file:

   `\IG Installation\tools\DISCIGIBridge\binaries\DISCIGIBridge.exe`

2. Launch VBS Blue IG.
3. Press **Tab** to open the VBS Blue IG Settings (on page 201) window.
4. Select **CIGI** in the open VBS Blue IG Settings (on page 201) panel.

5. Configure the Bridge application to listen to a specific IP and Port combination that DIS traffic is transmitted on. For more information, see Configuration (on the next page).

   a. In the open application, select the **CIGI** option.

   b. Set the Bridge IP and port settings for CIGI traffic to match the settings in the VBS Blue IG CIGI settings panel.

   c. Once the bridge has been configured, access the launch page by clicking **Launch** in the navigation bar.

      The application takes a few seconds to launch. Once the bridge has launched successfully, the text on the launch button changes to **Stop** .

      > **NOTE**
      > Information on the launched bridge mode is based on the configuration parameters.

   d. To stop the bridge, press the **Stop** button or close the application.

4.2.1.2 Bridge mode

Use this mode to combine an incoming CIGI Stream with the converted DIS > CIGI traffic.

Follow these steps:

1. Start the application directly by opening the .exe file:

   `\IG Installation\tools\DISCIGIBridge\binaries\DISCIGIBridge.exe`

2. Set up the CIGI Host to connect to the CIGI bridge specifying IP, Interface and Port settings as normal for connecting to VBS Blue IG,

3. In the Bridge application, use the settings for Bridge to Host communication in the application.

   For more information, see Configuration (on the next page).

4. Set up the application CIGI settings to communicate to the CIGI Host and the target VBS Blue IGinstance.

5. Launch VBS Blue IG.

6. Launch the CIGI host application.

7. Once the bridge has been configured, access the launch page by clicking **Launch** in the navigation bar.

   The application takes a few seconds to launch. Once the bridge has launched successfully, the text on the launch button changes to **Stop** .

   > **NOTE**
   > Information on the launched bridge mode is based on the configuration parameters.

8. To stop the bridge, press the **Stop** button or close the application.

4.2.2 Configuration

- CIGI Settings (below)
- DIS Settings (below)
- Entity Settings (on the next page)
- Entity Mappings (on the next page)
- Entity Detail (on the next page)

4.2.2.1 CIGI Settings

The CIGI Settings page can be accessed by selecting **CIGI** in the navigation bar. This page allows the user to specify the IP addresses of the CIGI Host (the source of incoming CIGI data) and the IG (the destination of outgoing CIGI data), along with the network ports to be used. In addition, settings are available to specify the maximum number of entities and designators included in the outgoing CIGI data, as well as the starting ID that should be assigned to each.

These settings are also available in the `\IG_Installation\DISCIGIBridge\binaries\Settings\CIGIConnection.xml` file.

For more information on each parameter, please refer to the Reference (on page 51) section of the documentation.

4.2.2.2 DIS Settings

The DIS Settings page can be accessed by selecting **DIS** in the navigation bar. This page contains settings for the incoming DIS data source. The IP address and port specifies the source of incoming DIS traffic, while the exercise, application and site ID is used to filter this traffic.

These settings are also available in the `\IG_Installation\DISCIGIBridge\binaries\Settings\DISConnection.xml` file.

For more information on each parameter, please refer to the Reference (on page 51) section of the documentation.

4.2.2.3 Entity Settings

The Entity Settings page can be accessed by selecting **Settings** in the **Entity** dropdown in the navigation bar. This page contains miscellaneous settings, including the CIGI type to display for DIS entities that do not have a mapping, the timeout for inactive entities, and the CIGI entities to display for various DIS detonations.

These settings are also available in the `\IG_Installation\DISCIGIBridge\binaries\Settings\EntitySettings.xml` file.

For more information on each parameter, please refer to the Reference (on the next page) section of the documentation.

4.2.2.4 Entity Mappings

The Entity Mappings page can be accessed by selecting **Mappings** in the **Entity** dropdown in the navigation bar. This page is used to provide custom mappings between DIS and CIGI entity types which override the default mappings contained in the `EntityMapping.xml` file.

To add a new mapping, click the **Add New +** button and a blank mapping appears at the bottom of the list. When entering a **DIS Type**, you can either enter a type ID directly or search by description and select the relevant type from the menu.

These settings are also available in the `\IG_Installation\DISCIGIBridge\binaries\Settings\UserEntityMapping.xml` file.

For more information on each parameter, please refer to the Reference (on the next page) section of the documentation.

4.2.2.5 Entity Detail

The Entity Detail page can be accessed by selecting **Detail** in the **Entity** dropdown in the navigation bar. This page is used to provide additional details for each DIS type, including the CIGI entity to display on detonation and information about the weapons and parts attached to the type.

To add details for an entity not already displayed, click the button **Add New +** and a blank entry appears at the bottom of the list. When entering a DIS Type, you can either enter a type ID directly or search by description and select the relevant type from the menu. This applies to the selection of a munition type of a weapon, as well.

To add weapons or parts to an entity, click the the button **Add New +** and a new weapon / part is added below those that already exist.

These settings are also available in the `\IG_Installation\DISCIGIBridge\binaries\Settings\UserEntityMapping.xml` file.

For more information on each parameter, please refer to the Reference (on the next page) section of the documentation.

4.2.3 Reference

- CIGI Settings (below)
- DIS Settings (on page 53)
- Entity > Settings (on page 53)
- Entity > Mappings (on page 53)
- Entity > Detail (on page 53)

4.2.3.1 CIGI Settings

Setting	Description	Allowed Values
Host IP	IP address of the CIGI host.	Valid IP Address
IG IP	IP address of the IG.	Valid IP Address
Host Interface IP	IP address of the CIGI host's network interface **NOTE** Use `IPADDRR_ANY` to listen on all interfaces.	Valid IP Address or `IPADDRR_ANY`
IG Interface IP	IP address of the IG network interface **NOTE** Use `IPADDRR_ANY` to listen on all interfaces.	Valid IP Address or `IPADDRR_ANY`
Multicast	Use if packets should be sent using multicast.	True / False

Ports

Setting	Description	Allowed Values
Bridge to Host	Port for network communications to the CIGI host.	Integer
Host to Bridge	Port for network communications from the CIGI host.	Integer
Bridge to IG	Port for network communications to the IG.	Integer
IG to Bridge	Port for network communications from the IG.	Integer

Entity IDs

Setting	Description	Allowed Values
Min Entity ID	Lowest CIGI entity ID to be used.	Integer, 0 - 65535
Entity Count	Maximum number of CIGI entities to be displayed.	Integer, 0 - 255

Designator IDs

Setting	Description	Allowed Values
Min Designator ID	Lowest CIGI designator ID to be used.	Integer, 0 - 65535
Designator Count	Maximum number of CIGI designators to be displayed.	Integer, 0 - 255

4.2.3.2 DIS Settings

Setting	Description	Allowed Values
Incoming IP	IP address of incoming DIS traffic.	Valid IP Address
Port	Port to listen for incoming DIS traffic.	Integer
Use Multicast?	Whether the DIS connection is using multicast.	True / False
Exercise ID	DIS Exercise ID	Integer, 0 - 255
Application ID	DIS Application ID	Integer, 0 - 65535
Site ID	DIS Site ID	Integer, 0 - 65535

4.2.3.3 Entity > Settings

Setting	Description	Allowed Values
Missing Entity Type	ID of default CIGI entity.	Integer, 0 - 65535
Entity Timeout	Timeout to remove inactive entities.	Integer, ≥ 0
Detonation Type	CIGI ID for given detonation type.	Integer, 0 - 65535

4.2.3.4 Entity > Mappings

Setting	Description	Allowed Values
DIS Type	DIS Type to be displayed as given CIGI Type.	Valid DIS Type
CIGI Type	CIGI Entity ID to be displayed for given DIS Type.	Integer, 0 - 65535

4.2.3.5 Entity > Detail

Setting	Description	Allowed Values
DIS Type	Type of DIS Type to apply details.	Valid DIS Type
Weapon Munition Type	DIS Type for weapon munitions.	Valid DIS Type

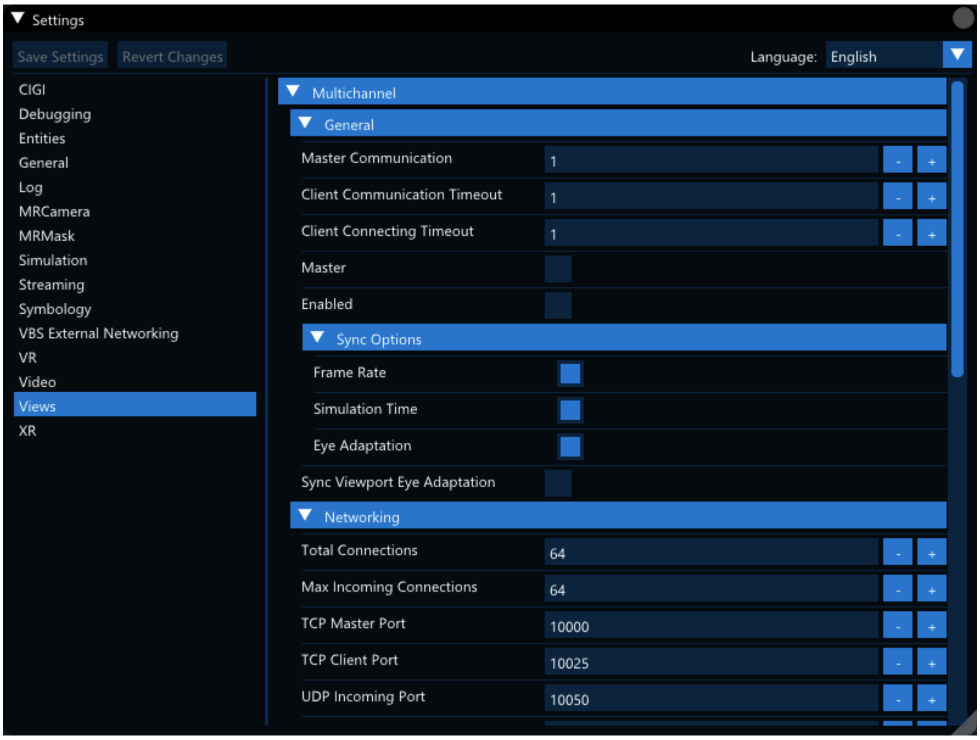
Setting	Description	Allowed Values
Weapon Effect	Select a weapon effect ID as configured in the following folder: `IG Installation \data\BlueProduct\mappings\Default\WeaponEffects.mapping`	Integer, 32-bit
Weapon Index	The weapon index as specified in the VBS Blue IG model.	Integer, 32-bit
Part ID	The part ID to articulate. Use the **Model Controller (on page 406)** window to inspect active models in the scene, their Articulated Sources and related Articulation IDs referencing individual parts of the model. Alternatively, export the desired model data using the **Content Library Generator (on page 388)** and inspect the generated XML files. The same information can be obtained from the CIGI Test Host Content Library. For more information, see CIGI Test Host - Overview (on page 21).	Integer, 32-bit
Part Class	A value from the SISO Standards DIS Enumerations document: 17.6.2 Articulated Part Type Class.	Integer, 32-bit
Part Metric	A value from the SISO Standards DIS Enumerations document: 17.6.1 Articulated Part Type Metric.	Integer, 32-bit

4.3 Views Settings

To synchronize multiple view channels, designate one of the VBS Blue IG Clients as the master client. IG Clients wait for a message from the master client before continuing processing. These clients communicate with the master through TCP and UDP network messages, while the master client communicates to all of its clients through multi-cast network messages.

Set up multi-channel views using either of the following methods:

- **Settings UI** - Press **Tab** to open the VBS Blue IG Settings (on page 201) and make adjustments, as required.



- **Edit XML file** - Open the `Views.xml` file, located in the following directory and edit, as required:

  `%LOCALAPPDATA%\Bohemia Interactive Simulations\`
  `VBS Blue IG\`*`version`*`\Settings\`

To setup multi-channel synchronization, **follow these steps**:

NOTE

There should be only one master client in any group of VBS Blue IG clients.

1. Enable the multichannel functionality for each IG instance. This should be done regardless of whether the instance is the master or client, and can be enabled using either of the following options:
   - Select **Views / Multichannel / General**, and click the **Enabled** checkbox.
   - Set the `<Enabled>` flag to `true` in the `Views.xml` file.
2. Designate a master client using one of the following options:
   - Select **Views / Multichannel / General**, and click the **Master** checkbox.
   - Launch the desired client with the `-master` command-line parameter.
   - On the target client computer, set the `<Master>` setting flag to `true` in the `Views.xml` file.
3. Determine and set a Multicast IP Address and port. All clients including the master client should have the same multicast address and port set.
4. Set the TCP address of the client instances to match the IP address of the PC where the master is running.

   > **NOTE**
   > In the field **Networking / Master Address**, the default value of the master instance is 127.0.0.1 (localhost). Although this default value can remain in the field, the actual IP address of the master client PC must be used in the client instances.

   All other ports only need to be adjusted, if they are already being used by another application.
5. Click **Save Settings**.
6. Restart VBS Blue IG.

Keep the following considerations in mind when using multi-channel synchronization:

- The master client and clients can be started and loaded independently.
- While VBS Blue IG loads, some clients may attempt to connect to the master and experience some time-outs. This stabilizes as soon as all clients finish loading.
- For optimal performance, synchronization works best in a low network latency environment, with VBS Blue IG running in full-screen mode and VSync turned on. In addition, the **fixed_screen_space_size_calc** option should be selected for all clients in the Video Settings (on page 340) menu.
- Master clients are not automatically found and determined, if no master can be found by a client, no deliberate synchronization occurs.

The tables below describe the type and purpose for each setting.

4.3.1 Views / Multichannel / General

Parameter (type)	Values
Master Communication **(uint8)** Max amount of time (seconds) a client waits for master render messages before it times out. Range 1 - 60	**Min / Max** 1 - 60 **Default** 1
Client Communication Timeout **(uint8)** Max amount of time (seconds) a master waits for client ready messages before it times out. Range 1 - 60	**Min / Max** 1 - 60 **Default** 1
Client Connecting Timeout **(uint16)** Max amount of time (seconds) a master waits for client first ready message before it times out. Range 1 - 300	**Min / Max** 1 - 300 **Default** 1
Master **(boolean)** Enables / Disables Master client mode	**Default** false
Enabled **(boolean)** Enables / Disables Multi-Channel functionality for this IG.	**Default** false

4.3.2 Views / Multichannel / General / Sync Options

Parameter (type)	Values
Frame Rate **(boolean)** When enabled, software based synchronization delays rendering of each frame until all members of the multichannel group are ready to render it. This effectively limits the frame rate to the lowest common denominator.	**Default** true
Simulation Time **(boolean)** When enabled, the current Simulation Time from the Master will be applied to the multichannel group.	**Default** true
Eye Adaptation **(boolean)** When enabled, Eye Adaptation values from the Master will be applied to the multichannel group.	**Default** true
Sync Viewport Eye Adaptation **(boolean)** If true, synchronizes eye adaptation between viewports on this VBS Blue IG instance without the overhead of IGMultiChannel networking.	**Default** false

4.3.3 Views / Multichannel / Networking

Parameter (type)	Values
Total Connections (uint16)	**Default** 64
Max Incoming Connections (uint16)	**Default** 64
TCP Master Port (uint16)	**Default** 10000
TCP Client Port (uint16)	**Default** 10025
UDP Incoming Port (uint16)	**Default** 10050
UDP Outgoing Port (uint16)	**Default** 10051
Multicast Port (uint16)	**Default** 10052
TCP Port Range (uint16)	**Default** 24
Multicast Address (string)	**Default** 225.0.0.100
Multicast NIC (string)	**Default** 0.0.0.0
Master Address (string)	**Default** 127.0.0.1

4.3.4 Views / Effects

Parameter (type)	Values
Apply Velocity Effect (boolean) Enables / Disables Velocity Effect on Camera	**Default** true

4.3.5 Views / Warping / Scalable

Parameter (type)	Values
Position Scale **(float64)** The multiplier to convert from the units defined in the configuration file, to meters. For example, 1.0 = meters, 0.001 = millimeters.	**Default** 0.01000
Warp Files **(string)** Defines the path to the warp file, ordered by search order. The warp data will attempt to load from the first path, and if not found, continue to the next, until all WarpFile paths are exhausted. Paths can either be absolute or relative to WarpComponent directory	**Default** `C:\Program Files\Scalable Display \DEI\LocalCalibration\ ScalableData.ol`

4.3.6 Views / Warping / Dome Projection

Parameter (type)	Values

Warp Files
(string)
Defines the path to the warp file, ordered by search order.
The warp data will attempt to load from the first path, and if not found, continue to the next, until all WarpFile paths are exhausted.
Paths can either be absolute or relative to WarpComponent directory
	Default
	`C:\DomeProjection\data\config.xml`
	`DomeProjection\config.xml`

Apply Frustum **(boolean)** If True, the view will have its frustum overridden by the frustum defined by the warp files.Default	**Default** true
Apply Position **(boolean)** If True, the view will have its position offset by the position defined by the warp files.	**Default** true
Apply Orientation **(boolean)** If True, the view will have its orientation offset by the orientation defined by the warp files.	**Default** true
PositionScale **(float64)** The multiplier to convert from the units defined in the configuration file, to meters. For example, 1.0 = meters, 0.001 = millimeters.	**Default** .001000

4.4 Virtual Reality / Mixed Reality Headsets

As an emerging technology, Virtual Reality applications and projects come with fundamental limitations. Operating within those limits, however, VBS Blue IG provides a robust environment for successfully running VR applications.

The Set Up Guidelines describe how to set up and implement HMDs (Head-Mounted Displays) / VR headsets with VBS Blue IG.

- VR / MR System Requirements (on page 64)
- HMD Device Compatibility (on page 66)
- Setting up VR / MR with VBS Blue IG (on page 72)
- MRMasking Component (on page 76)
- Troubleshooting VR / MR with VBS Blue IG (on page 85)

4.4.1 Definitions

Review this section to clarify the terminology used within the VBS Blue IG manual.

- **VR (Virtual Reality)**

  Virtual reality is commonly used to describe an immersive, three-dimensional, purely digital environment using head-mounted displays (HMDs) and hand-held controllers. The technology enables 360-degree views of this digital realm and the ability to interact directly with the media.

- **AR (Augmented Reality)**

  Augmented Reality brings virtual elements into the physical environment. An AR experience is one where the real world remains central, but is enhanced by adding digital media and details. A well-known example is Pokémon GO* which uses virtual overlaying.

- **MR (Mixed Reality)**

  Mixed Reality brings real elements into a virtual environment. In MR, you interact with and manipulate both physical and virtual items and environments. The technology provides immersion in the actual world around you while still interacting with a virtual environment using your own hands. For example, you can have one hand (or foot) in the real world, and the other in an imaginary world.

4.4.2 Programming Options

Review this section to understand the available programming options for VBS Blue IG and VR.

For Developers

Developers using VBS Blue IG typically add custom functionality by utilizing VBS IG SDK. Depending on the VR API version, custom functions enable the ability to query the orientation and position of the HMD and controllers, recenter the HMD, obtain eye separation, and handle input controllers, among other functions. For example, see MR Camera (on page 432) to see how integrating an API available in the VBS IG SDK enhances functionality of a component.

For Non-Developers

By using the built-in controls and view handling, non-developers can immediately visualize the scene in VR without any required programming. However, if the use case or project requires interaction beyond just simple viewing, the CIGI host needs to perform control feedback. The CIGI UI provides the ability to recenter the HMD and set up the view.

4.4.3 VR / MR System Requirements

Due to the evolving nature of the medium, system requirements for VR / MR (Mixed Reality) usage within VBS Blue IG are subject to change. However, the hardware and software recommendations below provide stable guidelines for running VR applications.

4.4.3.1 Hardware

- **Desktop Computer**

  The fastest Nvidia GPUs are recommended as a baseline. Currently, this would include the Nvidia 3080 / 3090 on the high end, followed by a 2080 or 1080Ti. Any GPU slower than the Nvidia 2070 or 1080 is not recommended.

  For the best results with any HMD model, users should follow the **Recommended** and **Optimum** specifications columns described in the VBS Blue IG System Requirements.

  > **NOTE**
  > Using a VR Headset requires high resolution stereo rendering, so hardware requirements are higher than running on a standard monitor at 60 / 75 / 90 FPS.

- **HMD** - See HMD Device Compatibility (on page 66).
- **Controller**

  a. The IG SDK supports out-of-box VR HMD controllers with the SteamVR / OpenVR runtime, which include the following controllers:
     - Meta / Oculus
     - HTC
     - Valve Index
     - Microsoft Mixed Reality Partners

  b. **Out-of-box Mixed Reality Masking Calibration** - Requires HTC Vive or Valve Index Controllers.

     Pairing USB dongles may be required depending on whether your HMD includes built-in receivers.

     For more information, see Setting up VR / MR with VBS Blue IG (on page 72).

- **Positional Tracking** - Given the wide variations in laser-based lighthouse tracking systems and other tracking cameras, providing specific recommendations for positional tracking is difficult. For recommended hardware, consult the documentation that accompanies your HMD.

4.4.3.2 Software

1. **Operating System** - Windows 10 Pro 64-bit (VBS Blue IG minimum requirement)
2. **Steam VR Runtime** - Download from the SteamVR Store.
3. **HMD software**- Download and install according to either option:
   a. **Windows MR** - VBS Blue IG supports Windows Mixed Reality for SteamVR software. Install from the SteamVR Store.
   b. **Varjo** - Install Varjo 3.9 Base Runtime from Varjo Downloads.
   c. **XTAL** – Install VRTool Online 3.2.0.135 or Standalone 3.2.0.69 from VRgineers.

4.4.4 HMD Device Compatibility

VBS Blue IG supports a variety of head-mounted displays (HMDs) for VR experiences that provide stereo rendering, head tracking, and positional tracking.

> **NOTE**
> Any OpenVR-compatible headset may work in your scenario. However, unlisted models have not been tested or guaranteed to work.

The following commercially available headsets are officially supported:

Manufacturer	Required Software	Model
Varjo	Varjo Base 3.9 or Newer	• Aero • VR-3 / XR-3 (foveated rendering and mixed reality pass through)
Oculus / Facebook / Meta	SteamVR with Oculus PC System Software	• Rift CV1 / S • Quest / Quest 2 / Quest Pro with PC Link Cable or Wifi
Microsoft Mixed Reality	SteamVR and Microsoft Mixed Reality for SteamVR	• HP Reverb / Reverb G2 • Samsung Odyssey / Odyssey+
VRGineers	VRTool Online 3.2.0.135 / Standalone 3.2.0.69 or newer	XTAL 3 / XTAL 3 MR (mixed reality pass through)
JVC	JVC HMD System Software and SteamVR	HMD-VS1W Augmented Reality
SA Photonics	Requires Head Mounted Tundra Tracker Skullcap, or otherwise additional tracking support to be provided with SDK.	SA-92/SA-147 Augmented Reality

Manufacturer	Required Software	Model
HTC	Viveport System Software and SteamVR Vive Streaming Software for supported Wireless Tether Headsets	• Vive • Vive Pro • Vive Pro 2 • Vive Pro Eye • Vive Focus 3 w/ Vive Business Streaming • Vive XR Elite w/ Vive Business Steaming
Valve	SteamVR	Index HMD

NOTE

HMD support is still in development. Performance, controls, and overall integration with many features is not complete. For more information, see Troubleshooting VR / MR with VBS Blue IG (on page 85).

4.4.5 Command Line Launch Options for HMDs

VBS Blue IG includes a wide range of command-line options for launching the application with supported headsets.

When running VBS Blue IG, use the `-hmd` command line to launch popular SteamVR-supported HMDs. A variation on this functionality is to append the command with a specific runtime, for example, `-hmd=openvr`. Additional parameters can be added to further customize the launched application.

Follow these syntax rules when using command line options to launch the application with a headset:

- Runtime Parameters (below) - The first identifier after the **=** must designate the runtime or specific headset support.

  Examples: `-hmd=varjo` or `-hmd=debug`.

- Sub-parameters (on the next page) - Additional sub-parameter commands can be included and are separated with an underscore (**_**).

  Example: `-hmd=varjo_mask`.

  In this case, the `_mask`sub-parameter provides mixed reality masking support for Varjo runtime HMDs.

VBS Blue IG supports three major VR runtimes (OpenVR, Varjo, XTAL) as well as additional special case headset modes that can be specified.

See the tables below for available command line parameters and their descriptions for launching with HMDs.

4.4.5.1 Runtime Parameters

Parameter	Description
`openvr`	The default display mode, intended for all the HMDs, unless otherwise better natively supported, and must support the OpenVR / SteamVR API. Typically installed with SteamVR on the Steam Platform.
`varjo`	Native Varjo runtime. Installed from Varjo Website.
`xtal`	Native XTAL runtime. Installed from VRGineers website.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td><code>viveproeye</code></td><td>OpenVR runtime with HTC support for Vive Pro Eye HMD.

NOTE
This option enables eye tracking to be collected through the IG SDK.</td></tr>
<tr><td><code>sapho92</code></td><td>IG window renders images designed for the SA Photonics SA-92 Augmented Reality HMD.

NOTE
HMD tracking must be incorporated through the IG SDK Plugin.</td></tr>
<tr><td><code>sapho147</code></td><td>IG window renders images designed for the SA Photonics SA-147 Augmented Reality HMD.

NOTE
HMD tracking must be incorporated through the IG SDK Plugin.</td></tr>
<tr><td><code>debug</code></td><td>Debug mode.
This option emulates stereo image rendering without any associated HMD runtime or hardware required.</td></tr>
<tr><td><code>debugfoveated</code></td><td>Debug foveated mode.
This option emulates quad image rendering without any associated HMD runtime or hardware required.</td></tr>
</table>

4.4.5.2 Sub-parameters

Once the runtime parameter after the `-hmd=` command is provided, additional sub-parameters can be used with headsets that support the functionality. Providing the parameter while using a non-supported headset will have no effect.

In the command line, sub-parameters are delimited by an underscore (_), and can be provided in any order.

Consult the following table for available sub-parameters.

Parameter	Description
`allviews`	Allows rendering of all the HMD incorporated Views into the main IG window if not otherwise already doing so. This sub-parameter is generally more useful for developers than for typical HMD users.

Parameter	Description
`blendcontrol`	Allows for reverse mixed Reality Masking to simultaneously composite on top of chroma key areas. Forcing areas to be virtual within masking areas regardless of being within the specified chroma color. **NOTE** Best used for masking in virtual objects in alignment with close by real world objects due to stereo camera convergence issues. *Example*: `-hmd=varjo_mask_keyblue_blendcontrol`
`depth`	Enables depth estimation for XR-3 hand occlusion.
`eye`	Enables eye tracking SDK usage for Varjo XR-3.
`hidecontroller`	Some headsets support 2 handed controllers. VBS Blue IG automatically render these for convenience unless you specify hiding the controller.
`keycolor`	Enables custom chroma keying color for Varjo XR-3 Mixed Reality. Color specified by Varjo Base Runtime.
`keyblue`	Enables chroma keying using built in Blue Chroma Profile for Varjo XR-3.
`keygreen`	Enables chroma keying using built in Green Chroma Profile for Varjo XR-3.
`markeron`	Initialize with active Marker Tracking (uses more CPU) for Varjo XR-3, while also enabling any provided Varjo Marker Mixed Reality meshes to automatically render when using the **mask** sub command.
`markeroff`	Initialize with deactivated Marker Tracking (using less CPU) for Varjo XR-3, while also enabling any provided Varjo Marker Mixed Reality meshes to automatically render when using the **mask** sub command.
`mask`	Enables Mixed Reality-based mask rendering for usage with Varjo XR-3 when using `varjo` runtime. The JVC HMD-VS1W when using the `openvr` runtime. The XTAL 3 MR when using the `xtal` runtime.
`maskd`	The same as the `mask` subcommand except it renders the masking into the depth buffer.
`nomenu`	For headsets that use the VBS Blue IG window for images. This removes any menu systems from rendering to the VBS Blue IG window, which will reduce display latency. Headsets currently supported are the SA Photonics SA-92 and SA-147 AR hmds.

Parameter	Description
`standing`	Specifies to use the standing mode tracking space, if headsets don't automatically launch into standing mode.
`vsync`	For headsets that use the VBS Blue IG window for images. Specify `vsync` to remove any undesired tearing. This slightly increases display latency. Headsets currently supported are the SA Photonics SA-92 and SA-147 AR HMDs.
`manualtools`	Specifies use of manual masking volume tools for use with Varjo XR HMD masking compositing modes.
`apprecenter`	Specifies use of application recenter based calibrations.
`meshcorrelation`	Uses a visual mesh to correlate where the HMD user will be placed relative to the tracking space.
`tomarker#`	Visual marker based calibration for Varjo XR-3, to be used in conjunction with calibrating in mask_meshcorrelation mode. Results in custom HMD recenter that will restore mask and user location to specified calibrated Varjo Marker ID#.
`nocamerainit`	Only Mixed Reality HMDs that support this will not initialize and turn on their MR camera hardware and require you to do it manually in the runtime software.

4.4.6 Setting up VR / MR with VBS Blue IG

Follow the procedures below to set up VBS Blue IG with an HMD.

- Basic Set Up (below)
- Best Practices for Set Up (on the next page)
- Alpha Masking (on page 75)

4.4.6.1 Basic Set Up

The basic method for using VBS Blue IG with an HMD requires setting specific command-line parameters.

Check that your HMD is properly set up and connected, following all HMD product-specific instructions:

Follow these steps:

1. Download the **Steam VR Runtime** from the SteamVR Store.
2. Run the Setup Tool, `SteamSetup.exe`.
3. Follow the Setup Tool instructions.
4. Follow any additional setup instructions specific to your individual HMD.

Launch VBS Blue IG with the command line option specified for your HMD:

- **All supported HMDs except Varjo** - Most HMDs can be configured to run using OpenVR / SteamVR which is launched using the `-hmd` or `-hmd=openvr` command line option.
- **Varjo** - Use the `-hmd=varjo` command line option to launch in Varjo Native runtime to enable advanced features like foveated rendering and mixed reality.

  Initialize the device with any of the additional options to enable Mixed Reality functionality:

  - `-hmd=varjo_mask` - Enables support for the 3D masking mesh.
  - `-hmd=varjo_keycolor` - Enables support for chroma key masking.
  - `-hmd=varjo_mask_keycolor` - Enables support for both 3D masking mesh and chroma key.

  For a full list of supported features, see Command Line Launch Options for HMDs (on page 68).
- **XTAL 8K** - Use either of the following modes with their command line options:
  - **OpenVR Mode**: `-hmd` or `-hmd=openvr`
  - **Native Mode**: `-hmd=xtal`
  - **Native Mode MR**: `-hmd=xtal_mask`

- **JVC HMD-VS1W** - Use either of the following modes with their command line options:
  - `-hmd=openvr`
  - `-hmd=openvr_mask`
- **SA Photonics SA-92/SA-147** - Use either of the following modes with their command line options:
  - `-hmd=sapho92`
  - `-hmd=sapho147`

For more information about command line parameters, see Command Line Launch Options for HMDs (on page 68).

To continue customizing your setup and optimize performance, use the Best Practices for Set Up (below) and Alpha Masking (on page 75) procedures.

4.4.6.2 Best Practices for Set Up

The following settings are recommended but not required. Users should test these settings within their specific environment and hardware setup to determine optimal results.

Follow these steps:

1. Open the VBS Blue IG Settings (on page 201) menu with the **Tab** key.
2. In the left pane, click **VideoSettings**.
   a. Click **Scene** - Move the **Biome Grass Detail** slider to the minimum setting to reduce grass detail.
   b. Click **Renderer** - **Multi-Projection Technology** - Select **MVR / SPS**

> **NOTE**
> **SPS** (Single Pass Stereo) is available with any installed Nvidia GPU (10-series or later) and **MVR** (Multi-View Rendering) for 20-series or better.

3. Click **Save Settings**.

Flight simulation use cases

For flight simulation use cases, adjust the following shadow settings to optimize frame rates:

1. If not already open, use the **Tab** key to open the VBS Blue IG Settings (on page 201) menu.
2. In the left pane, click **VideoSettings**.
3. Click **Renderer**.
4. For **Exterior Shadow Quality** / **Cascades** and **Interior Shadow Quality** / **Cascades** options, set to the minimum on the slider to turn off Dynamic Shadows.

5. Deselect **Biome Shadows**.

> **NOTE**
> This setting may already be unchecked by default.

6. Click **Save Settings**.

Integrating Camera MR

The Camera MR features provides options for integrating sourced imagery from cameras, videos, or images into a scene.

1. Open the Debug UI (on page 376) menu with the **~** key.
2. Click **MR Camera**.
3. Select from the options.

> **TIP**
> For more information about all settings available for the MR Camera option in the Debug UI window, see MR Camera (on page 432).

Frame Rate Recommendations

Setting the right frame rate is critical for reducing potential lag in VR graphics. Begin testing by using a lower framerate, the *reprojection target framerate*, which is typically half the refresh rate for a given headset. This allows the HMD to compensate with an extra frame to reduce any perceived latency.

Following a simple formula achieves the best results: *target fps = 1/2 x native refresh rate*.

Test your hardware by following these example HMDs:

HMD Model	Reprojection Target Framerate	Native Refresh Rate
Vive	45	90
Vive Pro	45	90
Rift S	40	80
Valve Index (multi-mode capability)	40/45/60/72	80/90/120/144

Additional Settings / Features in CIGI for HMD usage

- View attached symbol surfaces are automatically rendered on the HMD screen. The stereoscopic focal distance can be controlled by the **View Stereoscopic Focal Distance Component Control** packet (see the online ICD Reference (https://manuals.bisimulations.com/cigiicd/icd_23_2/index.html)).

- Position request packets for the view return the current HMD position, and not the view position as set by CIGI.
- To re-center the HMD position, use the **View Recenter HMD Component Control** packet (see the online ICD Reference (https://manuals.bisimulations.com/cigiicd/icd_23_2/index.html)).

4.4.6.3 Alpha Masking

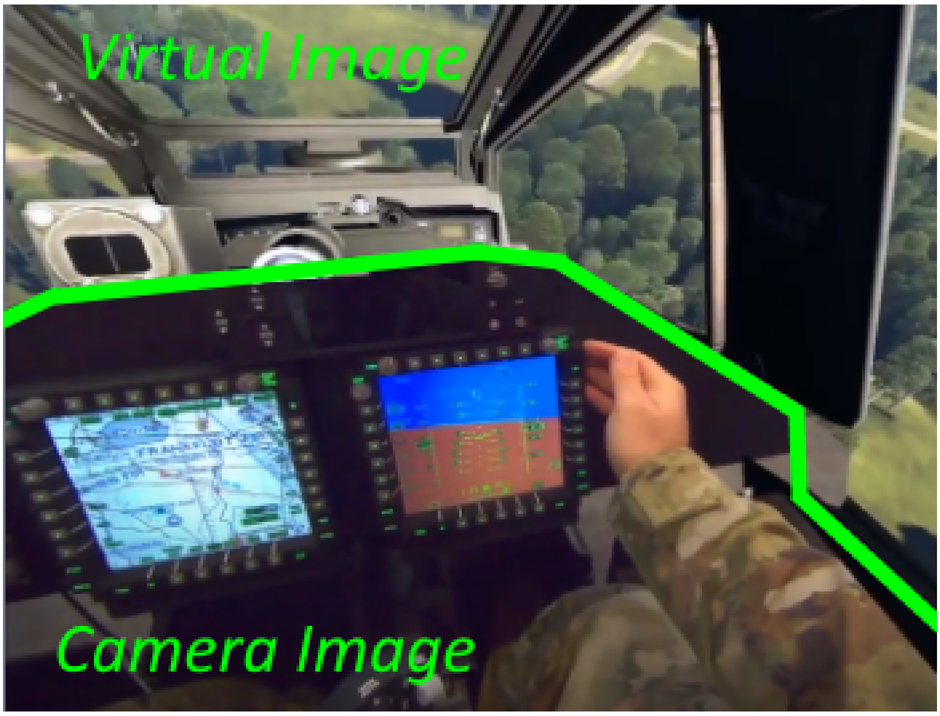
A new MRMasking Component is available in the Component Menu for alpha masking calibration support. For more information, see MRMasking Component (on the next page).

4.4.7 MRMasking Component

VBS Blue IG provides Alpha masking calibration for Mixed Reality (MR) HMDs using the **MRMasking** Component.

MR Masking calibration is primarily designed for cockpit-based simulations, allowing real world cockpit interface visuals to be merged into the application.

Image-3: A physical cockpit blended with a virtual representation of the aircraft interior and out-the-window view.



Alpha masking can be used in combination with chroma keying. This technique allows pass-through cut-out of more dynamic surfaces such as hands outside of the designated alpha masking area. Alpha masking can also be applied to override areas of the real world containing green chroma. For example, masking can override real world cockpit electronics using green visuals.

> **WARNING**
> Masking must exist inside the chroma key area.

Set up the MRMasking component by following the procedures in these topics:

- Enabling Component (on the next page)
- Creating Custom Masks (on page 78)

4.4.7.1 Enabling Component

To enable the MRMasking Sample Component, follow these steps:

1. Start VBS Blue IG with the `-hmd=openvr_mask` or `-hmd=varjo_mask` command line to turn on the general MR mode.

   For more information about command line parameters, see Command Line Launch Options for HMDs (on page 68).
2. Press **Tab** to open the VBS Blue IG Settings (on page 201).
3. Select **MRMask** in the left pane of the menu.
4. Click **Enabled** to activate the MRMask component.

For more information about configuring this component, see MRMask Settings (on page 281).

For more information about general HMD setup with VBS Blue IG, see Virtual Reality / Mixed Reality Headsets (on page 62).

4.4.7.2 Creating Custom Masks

VBS Blue IG allows integrators to develop custom MR masks that can be used with the Varjo mask-based MR functionality. These MR masks represent the geometry used to mask the portion of the virtual scene intended to be replaced with camera imagery.

4.4.7.2.1 Requirements for custom MR masks:

- Masks must be in position-only OBJ format, exportable from most modern modeling tools.
- Any number of masks are supported, but must have names ending with **Masking Mesh** and a designated number associated with the mask beyond the first mask.

  There are three example masks available in the following directory:

  `\IG_Installation\Systems\HMD\Assets\`

  You can either replace these or add more such as `Masking Mesh 4.obj`.
- During Calibration, you can toggle through available meshes found in the folder by pressing the HTV Vive or Valve Index controller **Pad** button.
- Mask meshes must share the same local origin as the `.p3d` model loaded into VBS Blue IG, for example, the virtual vehicle or aircraft used. This requirement ensures that the mask lines up correctly with the portion of the virtual cockpit to be masked.

4.4.7.2.2 Out-of-Box MRMask Support Component for cockpit alignment masking

Cockpit Masking with the MRMask component in VBS Blue IG is possible with two types of HMD device methods. Both methods currently require lighthouse tracking with a right handed SteamVR device to be paired and detected for calibration purposes. This is commonly done with a Right Handed Valve Index Controller or HTC Vive Wand Controllers. If HMDs do not have built in receivers for pairing controllers, you must provide additional an receiver dongle for pairing them.

Example launch options for out-of-box MRMasking Cockpit support include the following:

- `-hmd=openvr_mask_meshcorrelation` - AR HMD Device Support with SteamVR supported headsets like the JVC HMD-VS1W.
- `-hmd=varjo_mask_meshcorrelation` - Pass Through Camera Mixed Reality HMD Support with Varjo XR-3.
- `-hmd=xtal_mask_meshcorrelation` - Pass Through Camera Mixed Reality HMD Support with XTAL 3 MR.

Example Masking `Mesh.obj` mesh files are located in the following directory:

`\IG_Installation\Systems\HMD\Assets\`

These files contain simple position vertex index models that can be reproduced with most CAD tools. For optimal reasons, it is required that these mesh files have 3D position-only vertex and triangle index information. The origin space of these models must be equivalent to the actual aircraft model cockpit being loaded as a VBS Blue IG model. These models can be replaced with your own masking models.

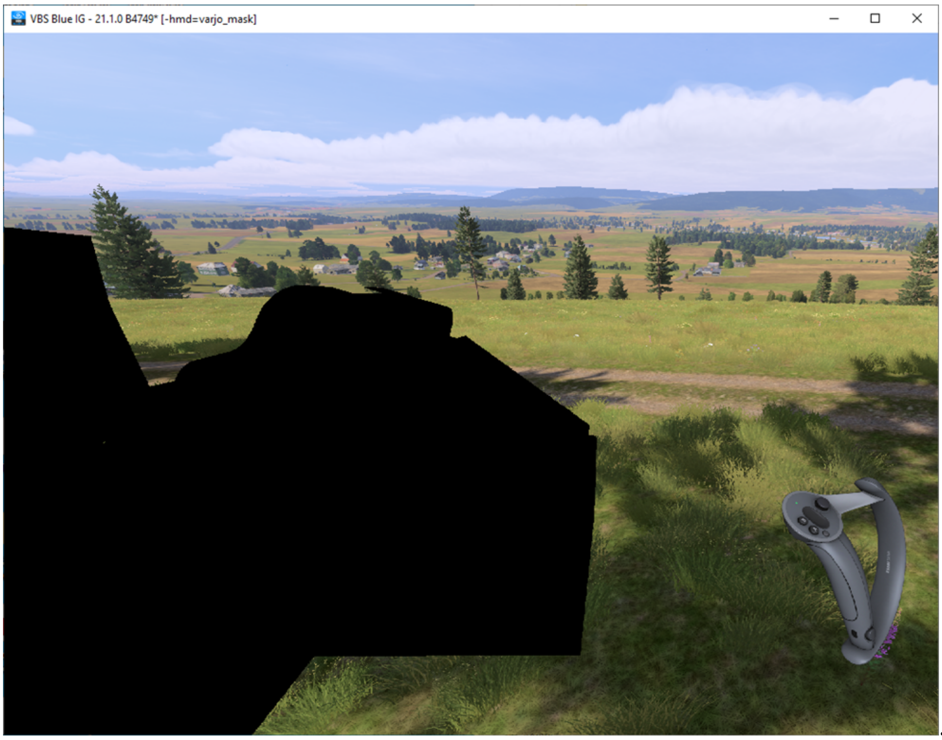
The following procedure describes a sample calibration using the provided example masking meshes and corresponding models.

> **NOTE**
> The sample model loaded in this procedure may instruct you to load through other means such as the CIGI protocol.

Follow these steps:

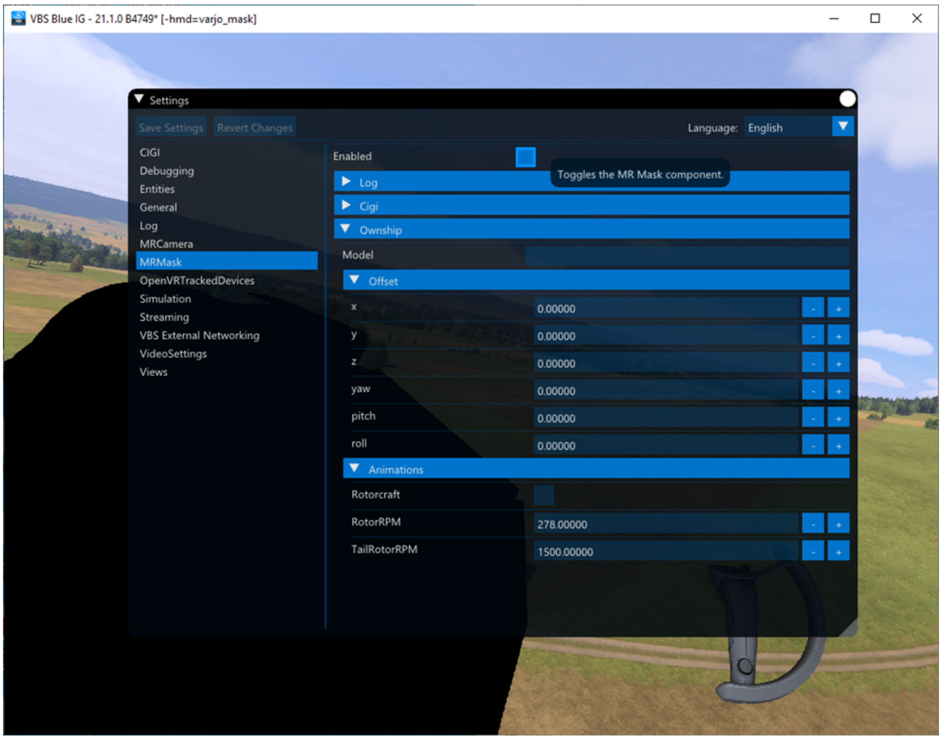
1. Launch VBS Blue IG.

   When the IG window opens, you should see a rendered black mask shape and a VR controller as shown in the image below.

2. Press **Tab** to open the Settings UI.
3. In the left panel, select MRMask.

4. Check the box to enable the component.



5. In the **Model** text box, type the string `vbs_demo_apache_vr_x` to render a sample helicopter into the scene with a mask insert.

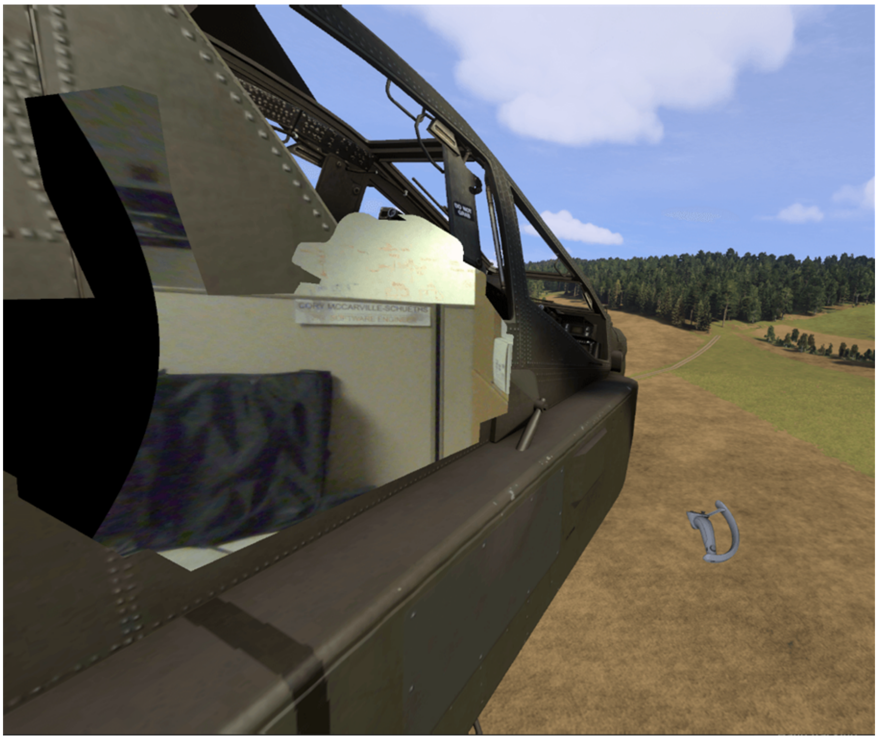
6. Click **Save Settings**.

7. Press **tab** to close the Settings menu.

   The mask for the cockpit should be rendered in black in the open window.

   If using an HMD with a mixed reality pass-through camera, real world images will appear inside the cockpit.

8. Enable 3D calibration mode, which can be tethered over USB or Paired Wirelessly with **steamvr** receivers. Wireless receivers are not always built into applicable headsets. Use one of the following controllers:

   - **HTC Vive controller** - Hold down the **menu** button .
   - **Valve Index controller** - Hold down the **B** button.

   Once enabled you will feel the controller rumble.

9. To adjust rendering of blended depth information, scroll thumb lightly on the controller touch pad you are using for calibration. This can assist in aligning the real world interface to the virtual mask.

   If using multiple masks, cycle through them by hard pressing the pad on the corresponding controller. For testing, be sure to switch this to the mask used in the pictures, since this is the mask that correlates to the sample helicopter.

10. Position and orient the model closer to you by doing the following:

    a. Hold the controller away from your body.

    b. Pull the trigger down.

    c. Pull and orient the model closer and then release the trigger to stop.

11. If the model is too far from you, pull and release the model a few times to bring it closer to the ideal position and orientation.

    Ideally, you will position and orient the mask mesh until you see it is aligned with the real world interfaces you want to see inside of it.

12. When finished positioning the mask, complete the set up according to your controller. This ensures that the calibration is saved, and can be reused without recalibrating again the next time you run the IG.

NOTE

Recalibration is required if the steamvr lighthouses are moved.

Select the applicable controller from the following:

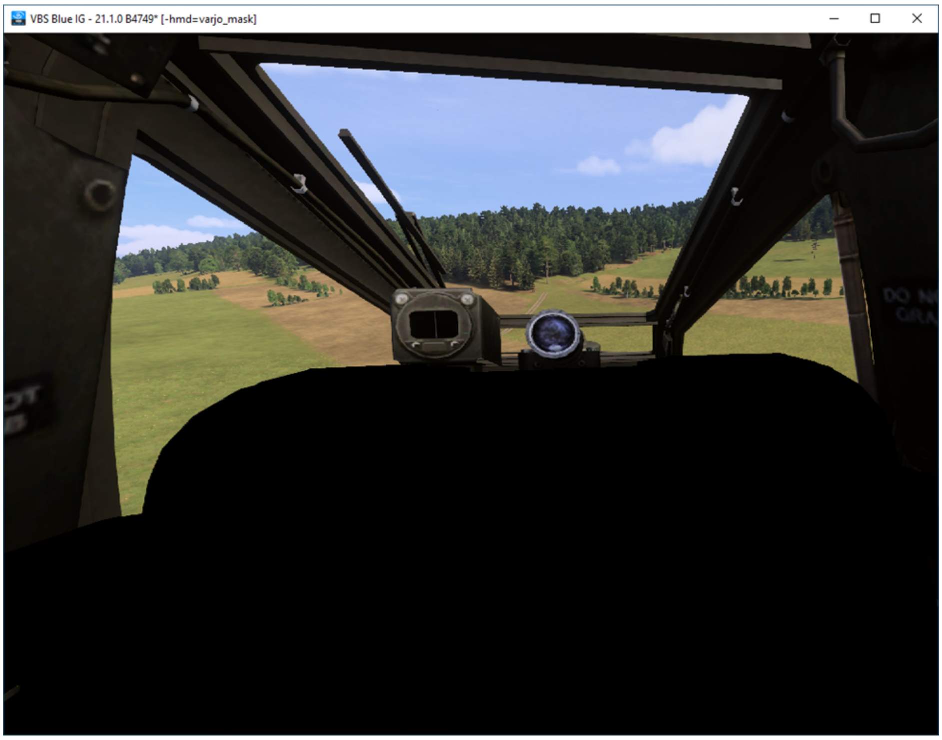
- **HTC Vive controller** - Hold and press the **menu** button.
- **Valve Index controller** - Hold and press the **B** button.

The test ship content and the masking should now move together spatially. The sample masking mesh `obj` file used here shares the same spatial origin as this Sample Helicopter VBS Content.

Design your masking mesh carefully to get the most effective blend of real world and virtual images with your desired VBS Content.

In your actual implementation, the real world masked instrumentation or interface may vary.

Once VBS Blue IG is loading your desired ship content over the network and drawing your desired `masking.obj` file, turn off the MRMask Sample Component to stop rendering this sample test ship.




DO NOT GRAB

4.4.8 Troubleshooting VR / MR with VBS Blue IG

- Verify that the SteamVR installation and VR HMD is working correctly with Basic Steam Software Installation features that accompany SteamVR RunTime (https://www.steamvr.com/en/).
- For all HMDs except Varjo, run VBS Blue IG with the `-hmd=openvr` command.
- When using Windows Mixed Reality, verify that the required client software is installed (see Software (on page 65)).
- When using a Varjo HMD, verify that the device displays Varjo HUB World correctly.

  Once verified, run VBS Blue IG with the `-hmd=varjo` command.
- Ensure that the drivers and firmware of the HMD, controllers, and tracking stations (if applicable) are up-to-date.
- For the Valve Index, Vive, and Vive Pro HMDs, small notification symbols, such as **!** or **i** may appear over the listed device icon in the SteamVR UI indicating that these require special attention.
- For Oculus HMDs, open the Oculus App in Windows and ensure that the connected headset is listed under **Devices** on the Oculus application. Confirm that there are no errors mentioned in the UI.

4.5 XR Training Platform: Overview

The XRTrainingPlatform (XRTP) is a VBS Blue IG component that correlates physical and virtual tracking spaces for Extended Reality (XR) with configuration options that persist upon restart. Specifically, features include the following:

- Calibrated physical tracking space that corresponds to a specific ownship entity.
- Ownship-class-specific offsets and scaling to match a virtual ownship interior to a physical space.
- Set up any arbitrary tracked objects, weapons, and configurable checklist-type signs and assign them to any shared OpenVR tracker, HMD, or controller with specific offsets. Virtual objects will be rendered and will move according to the tracker's real-world position with sub-mm accuracy.
- Provide weapon events (using ControllerInput) to a VBS4trigger tracked weapons.
- Create mixed-reality masks for Varjo XR and other mixed-reality headsets to allow "tracked" objects and weapons to be physical rather than represented by virtual models.
- Ability to share tracking space and all tracked objects between multiple IG clients.
- Assign and configure trackers to models and configure attributes in real time.

Example scenarios

- A virtual helicopter cabin with mounted turrets, and multiple crew (in VR) able to use any station.
- Ground infantry small arms weapons training.
- Mixed-reality aircrew training.
- VR vehicle training.

4.5.1 Using XRTP

Explore the topics below:

1. XRTP: Theory of Operation and Hierarchy (on page 88) - Provides high-level description of how XRTP works and some example use cases.
2. System Requirements for XRTP (on page 90) - Hardware and software needed for functionality.
3. XRTP Server Configuration (on page 91) - VBS4 host configuration.
4. XRTP Client Configuration (on page 93) - VBS Blue IG debug UI, SteamVR roomscale setup and tracker paring, and VBS Blue IG tracking space calibration.

5. XRTP Platform Configuration (on page 104) - Platform class-specific floor offsets and orientation of ownship respective to tracking space.
6. Tracker Configuration (on page 110) - Identifying paired trackers in VBS Blue IG, sample tracker coordinate systems, networked trackers.
7. Tracked Object Setup (on page 118) - Configuring tracked weapons, weapon attachments simple objects, checklists, and masks for users.

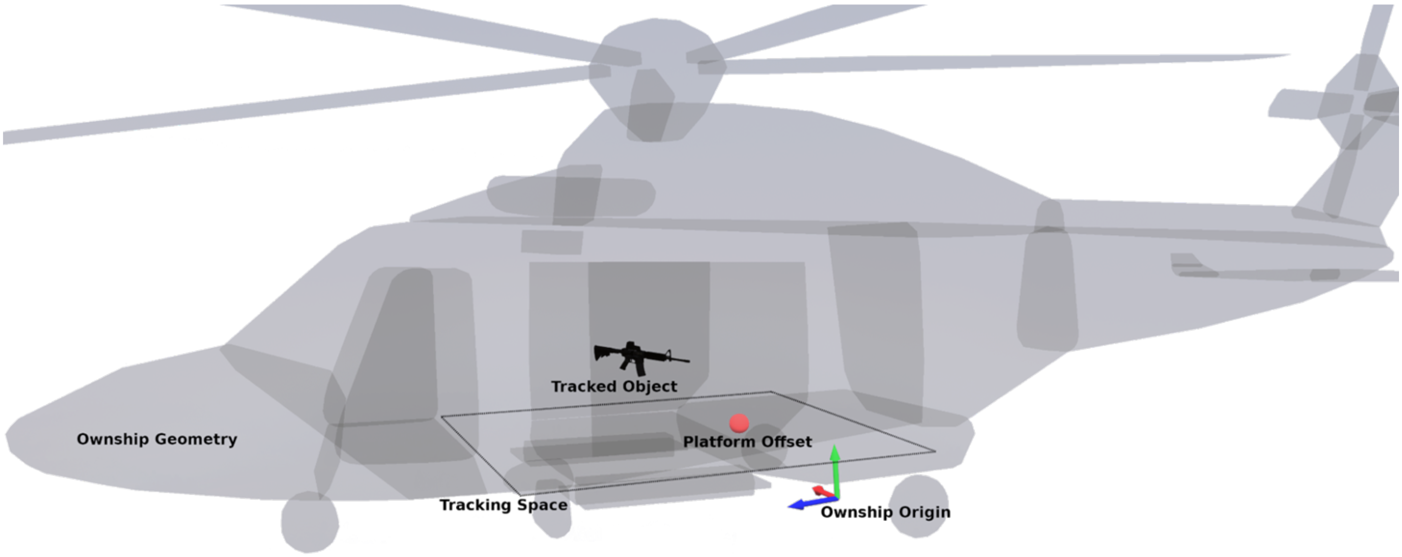
4.5.2 XRTP: Theory of Operation and Hierarchy

To properly configure XRTP requires understanding the spatial relationships between the virtual ownship (platform space) and physical tracking area (tracking space). There is a hierarchy in these spatial relationships:

- Ownship (on the next page)
- Platform Offset (on the next page)
- Tracking Space Offset (on the next page)
- Tracked Object (on the next page)

Image-4: Virtual 3D model.

Image-5: Hierarchy of spatial relationships within model.



4.5.2.1 Ownship

At the root of the hierarchy is the ownship. In VBS Blue IG, the **ownship model origin** is often at the base of the model, near its center.

4.5.2.2 Platform Offset

The direct child of the ownship is the platform offset, and corresponds to an abstract reference point for the tracking space origin.

For example, in the diagram above, standing / roomscale **tracking space** origin corresponds to a point on the physical floor, then the **platform offset** should correspond to a point along the cabin floor in the 3D model.

Since this offset can vary between different 3D models, the offset can be specified for each specific ownship type (**classname**) in the XRTP settings. The setting requires configuring and saving only once per model, and is invalidated only when the model is modified.

> **NOTE**
> Rotation and translations are completely configurable. However, in most cases, users only want to set the Y and Z translations for a given platform type.

4.5.2.3 Tracking Space Offset

The tracking space offset corresponds to a physical offset from the tracking universe origin. This offset converts real-world spatial transforms (for example, pose 2, 2 transforms from the headset, pucks and controllers) to virtual platform space, and is universal for all platform types.

Since this offset can be invalidated in various ways, such as switching VR hardware, moving lighthouses, performing room setup, VR runtime software update, a simple 3-point calibration procedure can calculate this offset automatically. For more information, see Tracker Configuration (on page 110).

4.5.2.4 Tracked Object

Tracked objects refer to virtual objects in platform space that can either track the movement of hardware tracking devices in tracking space, or be fixed at some arbitrary location in platform space. These objects have additional fixed transformation offsets, scale, and even attachments. For additional information, see Tracked Object Types (on page 115) and Tracked Object Setup (on page 118).

4.5.3 System Requirements for XRTP

- VBS Blue IG (VR client) and VBS4 host.

- Supported SteamVR lighthouse-tracked VR / MR headset on the client along with at least one hand controller (Vive wand or Valve Index Controller) for initial setup.
- (optional) Vive tracker, tundra tracker, or supported VR controller to be configured as weapon, checklist, or simple object.
- (optional) Vive / Tundra tracker (or other SteamVR-compatible generic tracker) for platform calibration.
- (optional) Mixed-Reality headset (for example, Varjo XR-3) for real-world mask support.
- (optional) Weapon, props and hardware to attach tracking devices.

4.5.4 XRTP Server Configuration

Use of the XRTP requires a VBS4 host configured either as a VBS4 host client or dedicated server. Additionally, a VBS4 component, **IGWeaponHost**, must be enabled.

The IGWeaponHost component allows a VBS4 **interop** host to receive weapon trigger events that fire virtual munitions within the simulation, as well as Laser Designators and their relative PRF codes.

> **NOTE**
> CIGI Hosts are not currently supported.

Configuration has two main parts:

- VBS Blue IG Setup (below)
- VBS4 Setup (on the next page)

4.5.4.1 VBS Blue IG Setup

Specific VBS Blue IG View Object XML files are required for setting up communication with a VBS4 server.

Follow these steps:

1. Open the following files included within the XRTP component folder at:

   `\IG_Installation\Components\XRTrainingPlatform\`

   - `views\host_view_objects\VRClient.xml`
   - `views\host_view_objects\VRClient2.xml`

2. **Optional**: You may modify the View ID number, if needed. However, the `ManualAttach` field must be set to **true**.

3. Copy both XML files to the VBS4 host installation at the following path:

   `VBS_Installation\Settings\CIGI\Views\`

4.5.4.2 VBS4 Setup

In order for XRTP to communicate with VBS4, the IGWeaponHost component must be enabled within VBS4:

Follow these steps:

1. **Option 1:** In the VBS Launcher, add the command line parameter `-igweaponhost`.
2. **Option 2:**
   a. Open the settings file `IGWeaponHost.xml` settings file located in:
      `%LOCALAPPDATA%\VBS4\Settings\`.
   b. Set the **Enabled** variable to **true**.

> **NOTE**
> If a server or VWS is being used in the scenario, the IGWeaponHost should only be enabled on the server, and disabled in all other VBS4 instances.

4.5.5 XRTP Client Configuration

We recommend performing SteamVR room scale setup before calibrating within VBS Blue IG.

The two user interfaces this guide uses with VBS Blue IG are the DebugUI and Settings UI.

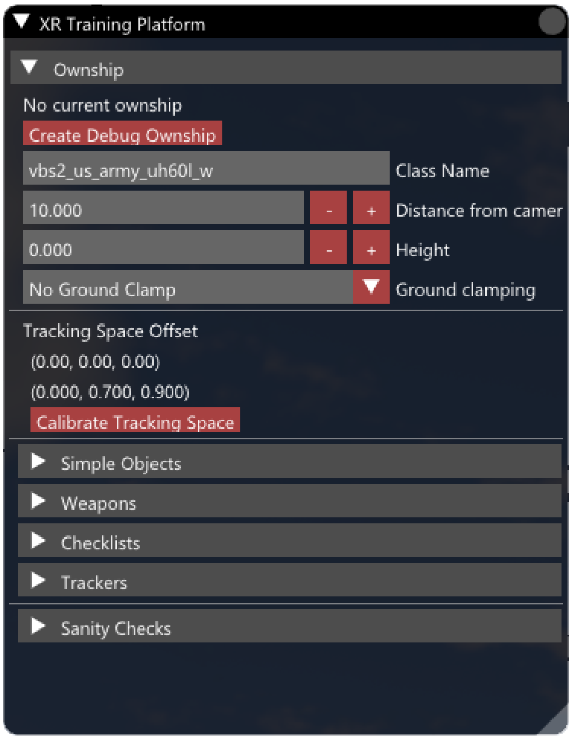
- XR Training Platform Debug UI (below)
- XR Training Platform Settings UI (on the next page)

Set up trackers pairing, roomscale, and calibration:

- SteamVR Tracker Pairing (on page 95)
- SteamVR Roomscale Setup (on page 98)
- Tracking Space Calibration (on page 99)

4.5.5.1 XR Training Platform Debug UI

The XRTP DebugUI is accessed by selecting the ~ key (by default), then selecting **XR Training Platform**.



The Debug UI is used for creating any *debug ownship* without having a host connected as well as performing Tracking Space calibration. It has functions to assign any weapon to any slot, create debug arrows to visualize offsets, test munition / fire effects, toggle weapon lasers, and debug the state of the component.

> **NOTE**
> None of the modifications made in the debug UI persist after VBS Blue IG is shutdown.

4.5.5.2 XR Training Platform Settings UI

The XRTP Settings UI is accessed by pressing the Tab key (by default) and selecting XR on the left pane. From here you can make any persistent settings as well as configure offsets for the tracking space, various platforms and weapons.

Setting Section	Description
Enabled	Toggles whether the XRTP component is active. If you're reading this document, you probably want this checked.
TrackingSpace	Tracking space offset from the SteamVR tracking universe origin. These values are usually calculated from the tracking space calibration procedure, but can be manually adjusted here.
Lasers	Global laser settings for any attached weapon laser.
Platform	Platform global settings.
Platform/Platforms	Platform-specific array.
SimpleObjects	Simple objects array.
WeaponObjects	Weapon-specific array.
PlatformGunnery	Weapon slot array.
CheckLists	Checklist array.

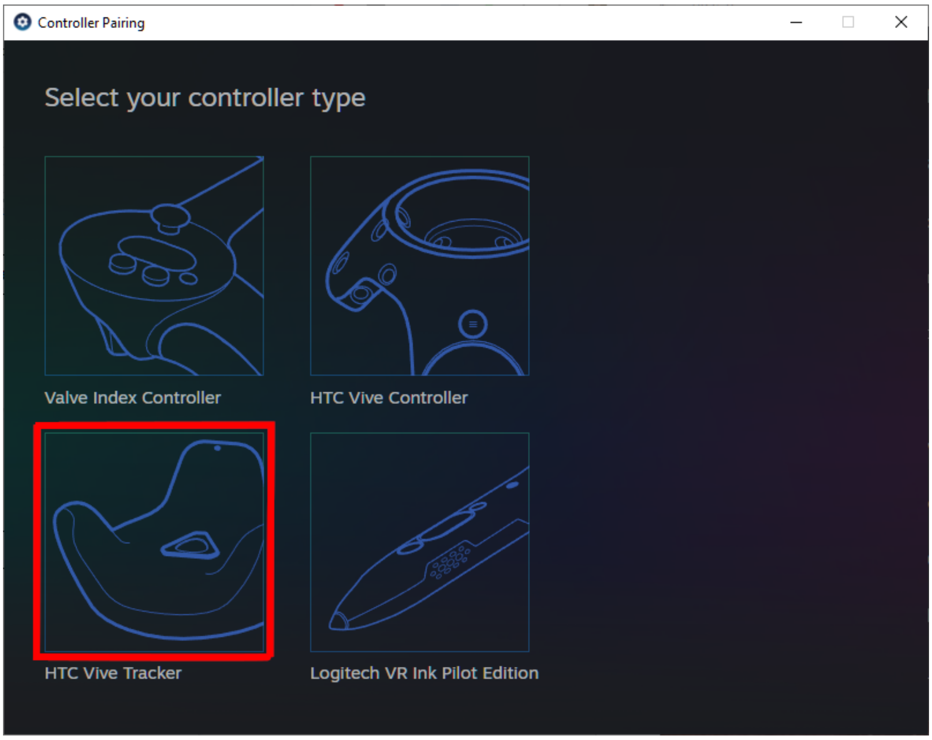
Setting Section	Description
Masks	Masks array.

4.5.5.3 SteamVR Tracker Pairing

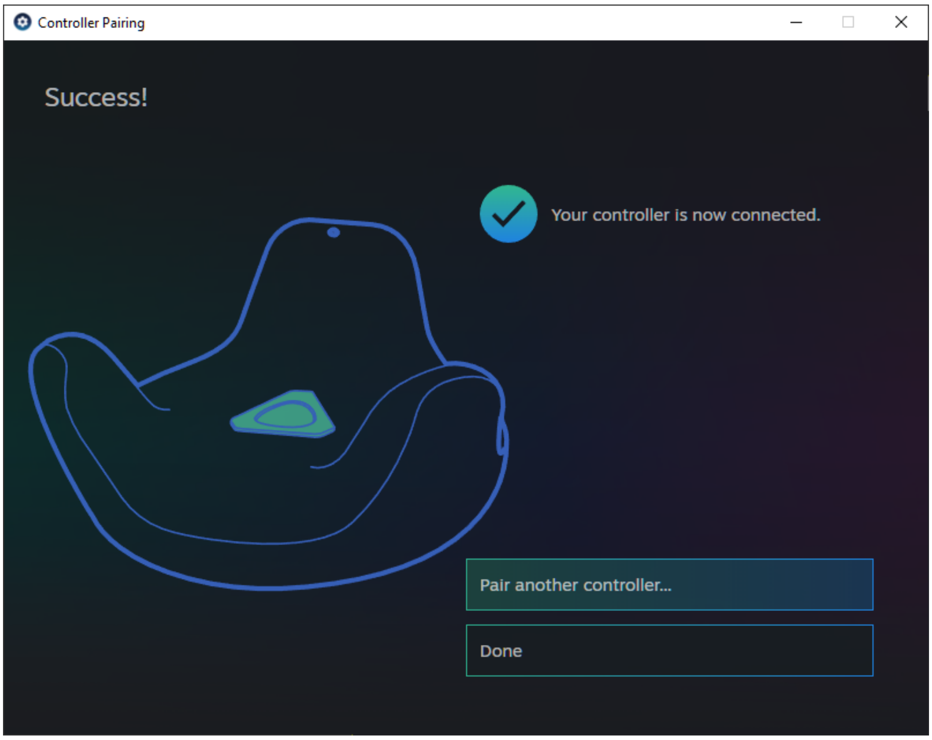
This only needs to be performed once per tracker.

Follow these steps:

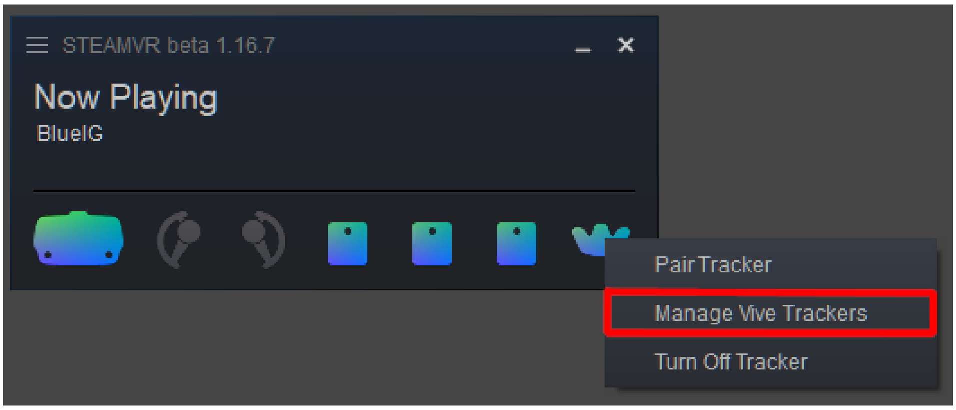
1. Install SteamVR runtime on each client PC. Ensure they are running the same version. Run SteamVR once to ensure that HMD hardware works.
2. Pair puck trackers with SteamVR by right-clicking on the SteamVR UI, selecting Devices and Pair Controller. Regardless of tracker type, select the HTC Vive Tracker icon and follow the instructions to pair the device to the client system. This only needs to be performed on one client system.



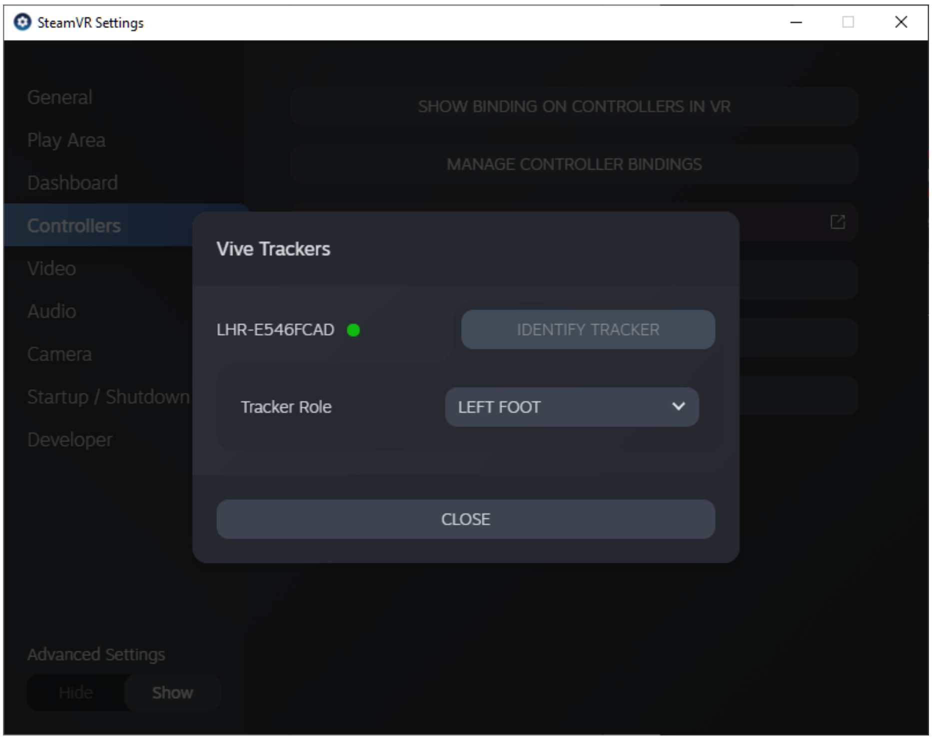
3. Confirm pairing.



4. Once each tracker has been paired, we need to reconfigure its role. Right click on each tracker's icon in SteamVR and select **Manage Vive Trackers** (or equivalent).



5. Change the tracker role to any appendage other than "Held in Hand". The uniqueness or relevance of what is chosen to the tracker's intended purpose is unimportant, except that "Hand" roles have a different behavior and orientation.



6. Close the dialog.

> **NOTE**
> If you're having trouble connecting to the tracker, it may require a wireless dongle to be plugged in. Usually only two wireless VR devices can be paired to the headset at any time; if the Index or Vive controllers are paired to the headset, then SteamVR requires a separate dongle to connect to additional devices.

For troubleshooting HTC Vive tracker connections: https://www.vive.com/us/support/wireless-tracker/category_howto/using-the-dongle.html.

For troubleshooting Tundra Tracker connections: https://docs.tundra-labs.com/.

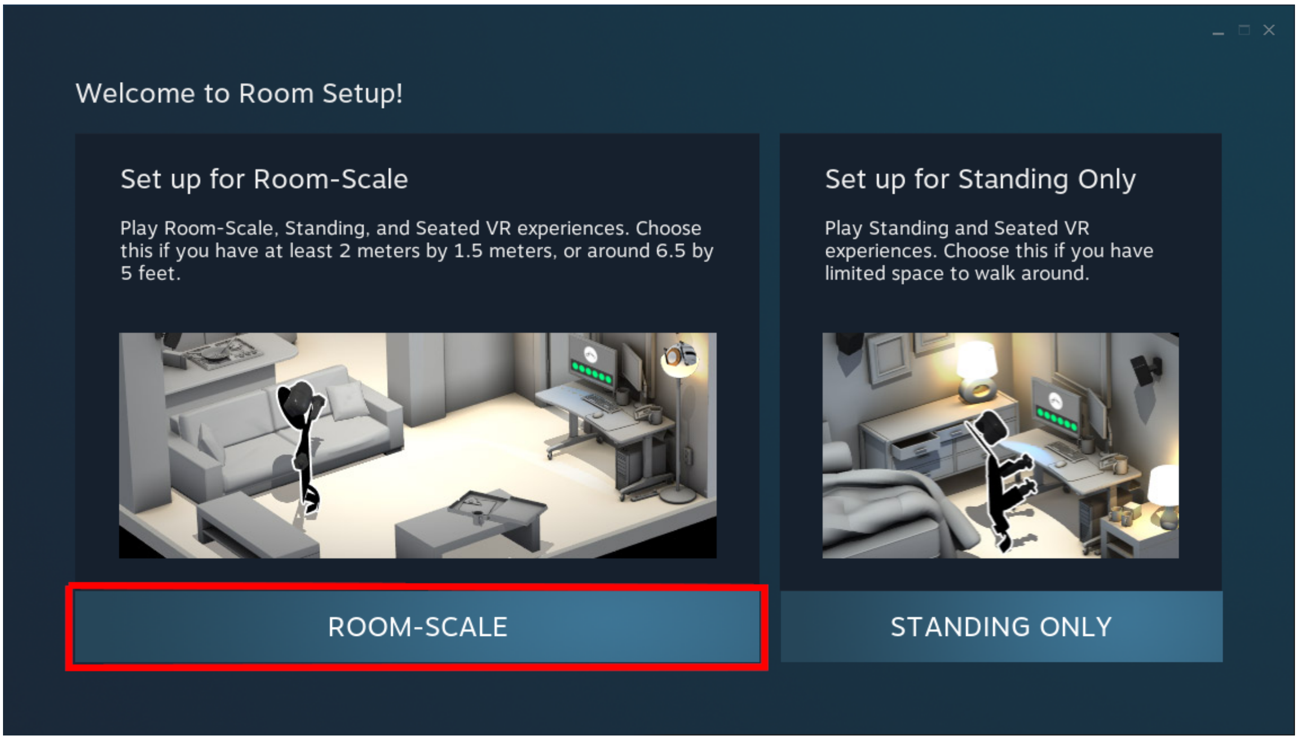
4.5.5.4 SteamVR Roomscale Setup

Roomscale setup needs to be performed whenever the lighthouses are moved or headset is changed.

Follow these steps:

1. Run SteamVR. If this is the first time steamVR is run, this dialog appears automatically. Otherwise, right-click on the SteamVR window and select **Perform Room Setup**.

   This dialog appears:



2. Select **ROOM-SCALE**.
3. Follow the on-screen instructions. Ensure that the chaperone boundary encompasses the entire physical area you wish to simulate.

> **NOTE**
> This part requires a supported hand controller such the Vive wand or Index Controllers.

4.5.5.5 Tracking Space Calibration

The tracking space calibration aligns the position and rotation of the virtual platform to the physical tracking space.

This procedure needs to be performed whenever Room Scale calibration is performed, or if SteamVR calibration has changed otherwise.

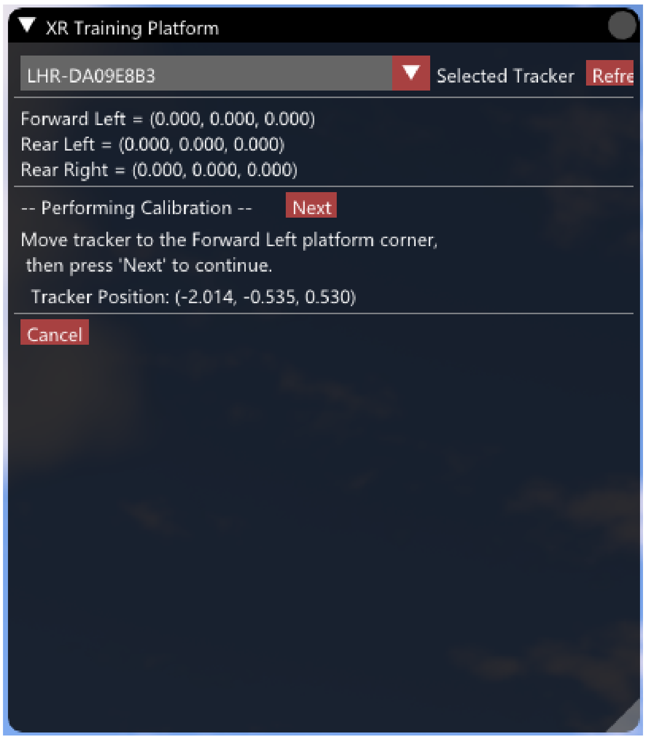
Follow these steps:

1. In VBS Blue IG, open the Debug UI menu using the **`~** key.
2. Select **XR Training Platform** from the debug UI menu.

3. Select **Calibrate Tracking Space**.

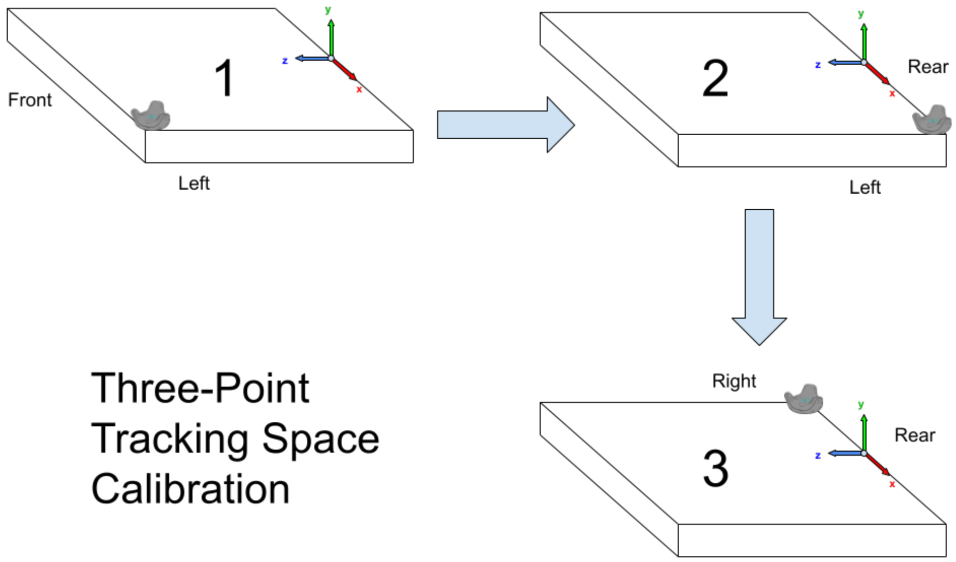
4. Select the tracker you'd like to use for calibration from the combo menu.

> **NOTE**
> The tracker name is unique to the physical hardware tracker.



a. If a tracker is not active, make sure it is turned on and paired with SteamVR.
b. Once paired and turned on, hit the Refresh button and it should be added to this combo list.

5. Perform the 3-point calibration procedure by placing the puck in 3 different positions:



   a. Move the puck to the front-left floor of the physical space (corresponds to forward-port side of ownship). Select **Next**.
   b. Move the puck to the rear-left floor of the physical space (corresponds to aft-port side of ownship). Select **Next**.
   c. Move the puck to the rear-right floor of the physical space (corresponds to aft-starboard side of ownship). Select **Next**.

6. If the 3 positions are approximately orthogonal, the orientation and position of the new tracking space offset is calculated.

The confidence is based on how close the forward-left and rear-right are to a right angle. 1.00 = perfect 90 degree angle

If the confidence is too low (<0.90), it won't calculate the transform and step 5 must be repeated.

a. Select **Save** to store the calculated tracking space offset but keep this current debug UI view open.

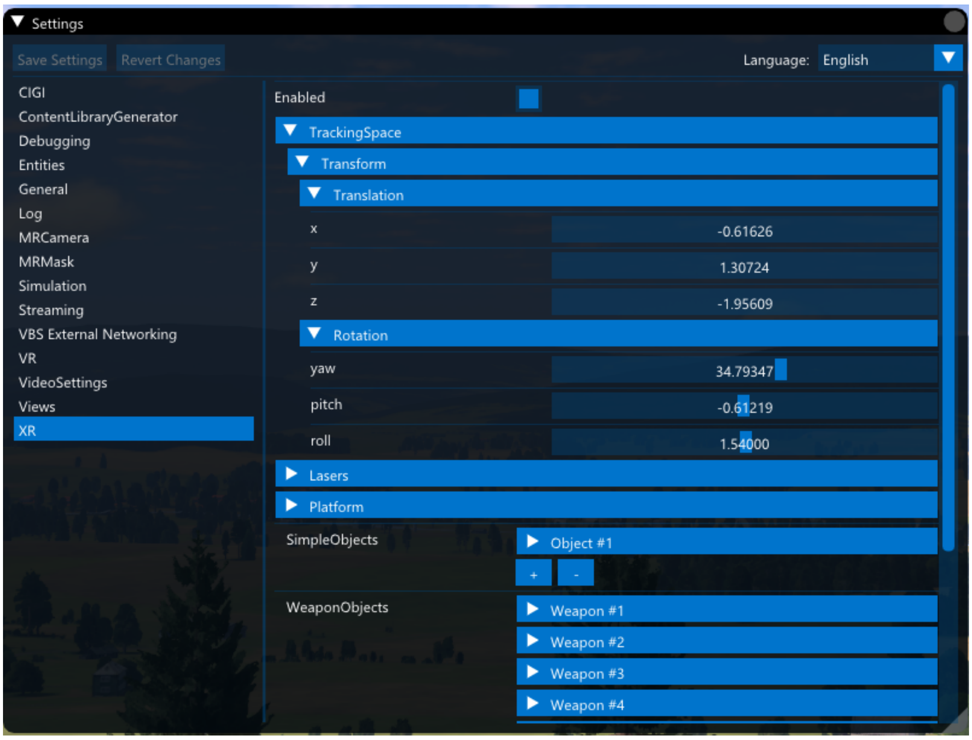
b. Select **Save and Finish** to store the calculated offset and close the calibration dialog.

c. Select **Restart Calibration** to not store this offset and repeat step 5.

d. Select **Cancel** to close the calibration dialog.

7. Press **Tab** to open the Settings UI.

8. Select **XR** > **TrackingSpace** > **Transform** to show the saved and stored calibration.



4.5.6 XRTP Platform Configuration

Some platform types require additional offsets in order to align the physical floor to the virtual floor of the vehicle content 3D model.

- Platform Settings (below)
- Adding New Platform Type (below)
- Debug Ownship (on page 108)

4.5.6.1 Platform Settings

Platform settings are found under the XR settings UI.

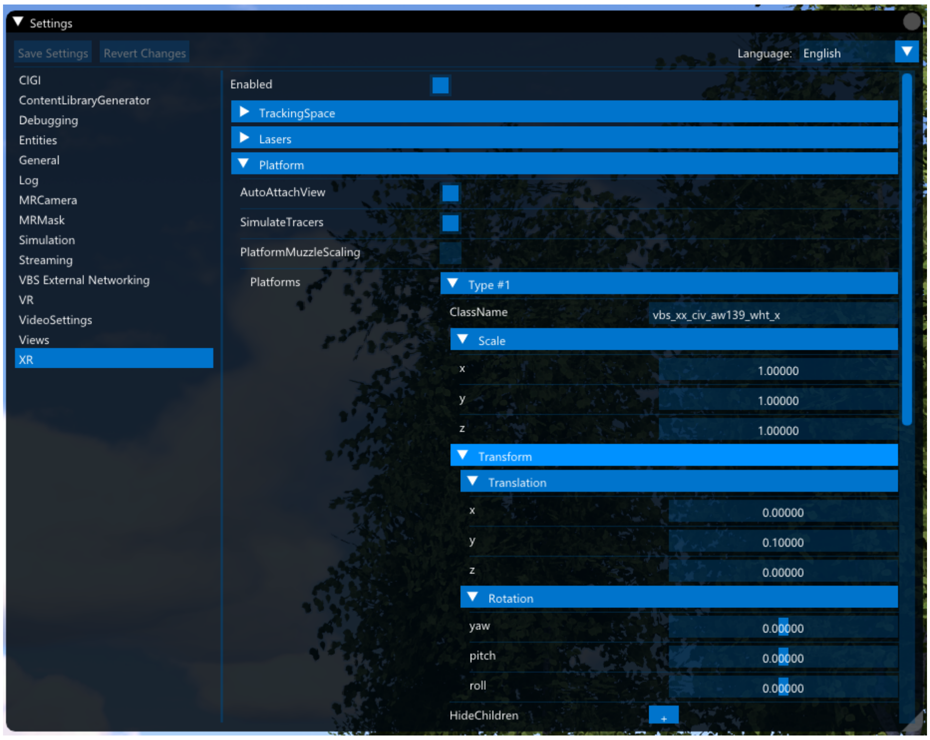
Setting Section	Description
AutoAttachView	If enabled, this automatically attaches the VBS Blue IG main view ID to the platform + platform-specific offset.
SimulateTracers	If enabled, this simulates munition tracers from the Host. In most cases, you'll want to leave this enabled.
PlatformMuzzleScaling	This scales the weapon muzzle offsets based on the platform scaling (below). Depending on host settings, this can break aim point correlation on IG Weapon hosts when firing munitions. For advanced users only.
Platforms (array)	An array of offsets. See Adding New Platform Type for more detail.

4.5.6.2 Adding New Platform Type

Follow these steps:

1. Start VBS Blue IG without `-hmd` enabled.
2. Press **Tab** to open the Settings UI.
3. Add the new platform:
   a. Expand **Platform**.
   b. Select the **+** button.
   c. Set the platform type.

4. Add the VBS classname (CfgVehicles entry) of the platform type. This ensures that these platform-specific offsets are applied whenever the host attaches the special IG view object to this particular entity.



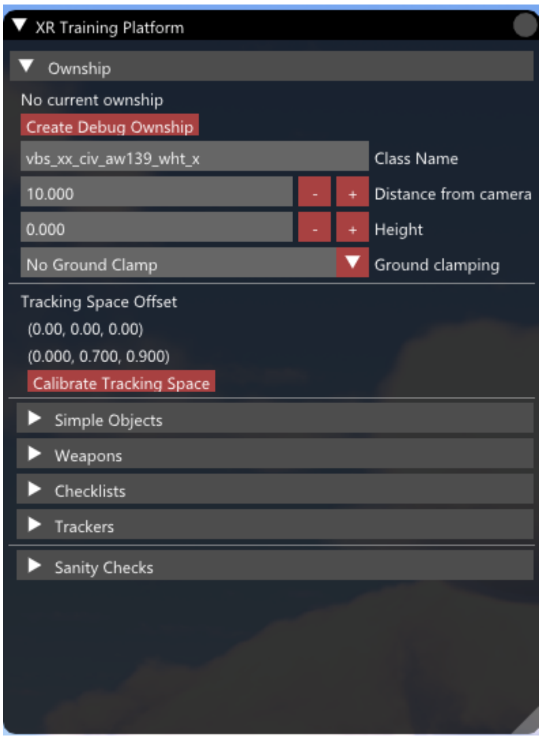
In our example, we use `vbs_xx_civ_aw139_wht_x`.

5. Click **Save Settings**, but keep this settings window open with the platform entry.

6. Use the debug menu to spawn this platform by entering this class name and pressing **Create Debug Ownship**.

7. Open the debug UI with the **~** key.

8. Select **XR Training Platform**.



Assuming the class name is valid, an entity of the given type should spawn.

The IG view should jump to a similar view as shown in this screenshot:

If the view does not jump to this object, ensure that **AutoAttachView** is enabled. Click **Delete Ownship** and then repeat step 5.

9. Manually adjust the offset within XR settings, under **Platform > Platforms > Type#N** (corresponding to the class name of the object just spawned). Ideally, the view should "rest" on the floor of the virtual platform since this corresponds to the physical floor in the real platform.

Both translation and rotation offsets can be modified, which update in realtime. However, it is not advisable to adjust rotation except in rare circumstances, and rely on Tracking Space calibration to select default orientation.

Typically, only the translation Y (+height) and Z (+forward) need to be adjusted in this step.

> **NOTE**
>
> Press **CTRL+click** on the setting edit boxes to fine-tune adjustments.

Scale can also be adjusted so that the virtual floor lines up with a physical tracking space. However, this is recommended to be left at 1,1,1 and only intended for advanced users since this breaks realistic scaling of real-world objects.

10. Press **Save Settings** when the offset appears to be in the middle of the virtual floor.
11. Restart VBS Blue IG in `-hmd` mode.

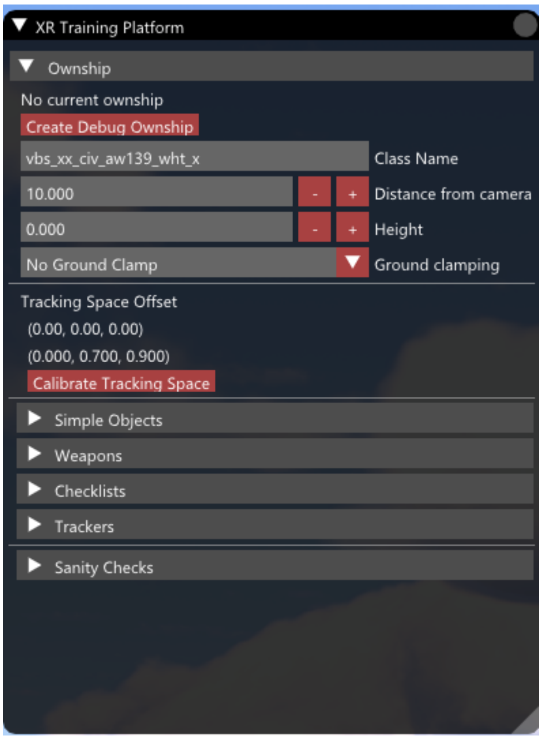
    Spawning the ownship via XRTP DebugUI or through the **interop** host (VBS4 + IG View Object) results in the HMD view being attached to the ownship such that the virtual floor of the ownship model correlates to the physical floor.

4.5.6.3 Debug Ownship

The debug ownship is a special object generated in the XRTP DebugUI. It can be used to test out view, platform and tracker offsets without having to manually spawn specific platforms in the interop host, or even be connected to a host.

Follow these steps:

1. Open the debug UI with the ~ key.
2. Select **XR Training Platform**.
3. Click the **Ownship** dropdown menu.



4. Click the **Create Debug Ownship** button.

   Configure and spawn ownship classname and starting position in this panel.

   Once created, the **Create Debug Ownship** button, classname , and positional configuration fields are replaced by ownship velocity (world velocity) and angular velocity.

NOTE

- The debug ownship can only be spawned if there is no active ownship, which prevents any conflicts while the interop host's ownship is driving the client.
- The debug ownship resides only on local client instances. Any interop host and other IG clients are not able to see or interact with it.
- If a debug ownship is active and the interop host creates a new ownship, the debug ownship is removed and replaced with the host ownship.
- Extra logic is applied to debug ownship helicopters, such as enabling rotor animations and opening rear doors.

4.5.7 Tracker Configuration

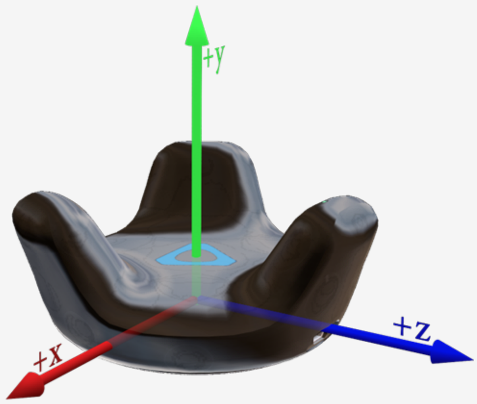
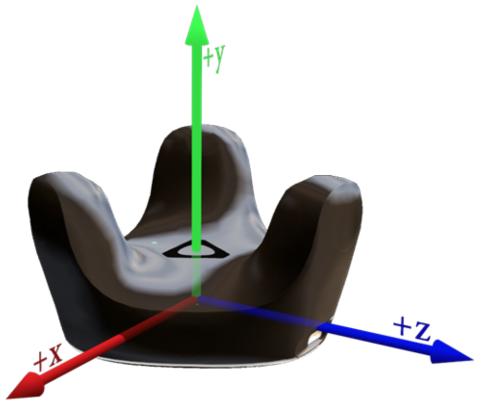
Configure trackers by following the procedures below.

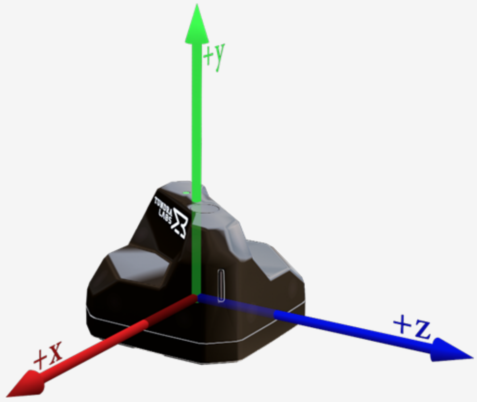
- Tracker Coordinate System (below)
- Tracker Name (on the next page)

4.5.7.1 Tracker Coordinate System

When determining offsets for tracked objects, it is important to know the coordinate system as well as the hierarchy for each object type.

The origin and rotation can differ between type of tracker as well as SteamVR tracker role. The following table describes the default origins and axes for the common trackers in non-hand roles.

Device	Description	Tracker Default
HTC Vive v1 **HTC Vive v2**	On both the HTC Vive v1 and v2 tracker, the LED faces forward along the +Z axis along with the charging port. The X,Z origin runs directly through the middle of the center button and the ¼” threaded hole on the bottom, and the origin’s Y (height) is near the top of the charging port.	
HTC Vive v3	Charging port is forward along the +Z axis and origin's X,Z lines up with the center button and ¼" threaded hole on the bottom, while the origin's Y is near the top of the USB-C charging port. Unlike the previous versions, the LED faces the rear.	

Device	Description	Tracker Default
Tundra Labs Tracker	The Tundra Tracker's USB-C port, button and indicator LEDs are oriented at a 45 angle to its base. The USB port is facing forward right, while the LED is facing rear-aft. The origin's X,Z lines up with the ¼" threaded hole on the bottom, while the origin's Y is near the baseplate.	

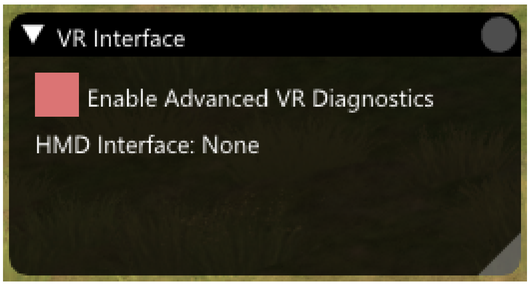
4.5.7.2 Tracker Name

Hardware tracker devices have unique names strings (often "LHR-XXXXXXXX") that can be specified in each of the Tracker fields for the tracked object types. This field is found under `SimpleObjects`, `PlatformGunnery`, `Checklists`, and `Masks`.

Find the tracker name using the procedure below.

Follow these steps:

1. Press backquote (`) / tilde (~) on the keyboard to open the Debug UI.
2. Select **VR Interface** to open the VR Interface debug panel.
3. On the VR Interface debug window, check **Enable Advanced VR Diagnostics** to enable new additional options in the Debug UI.



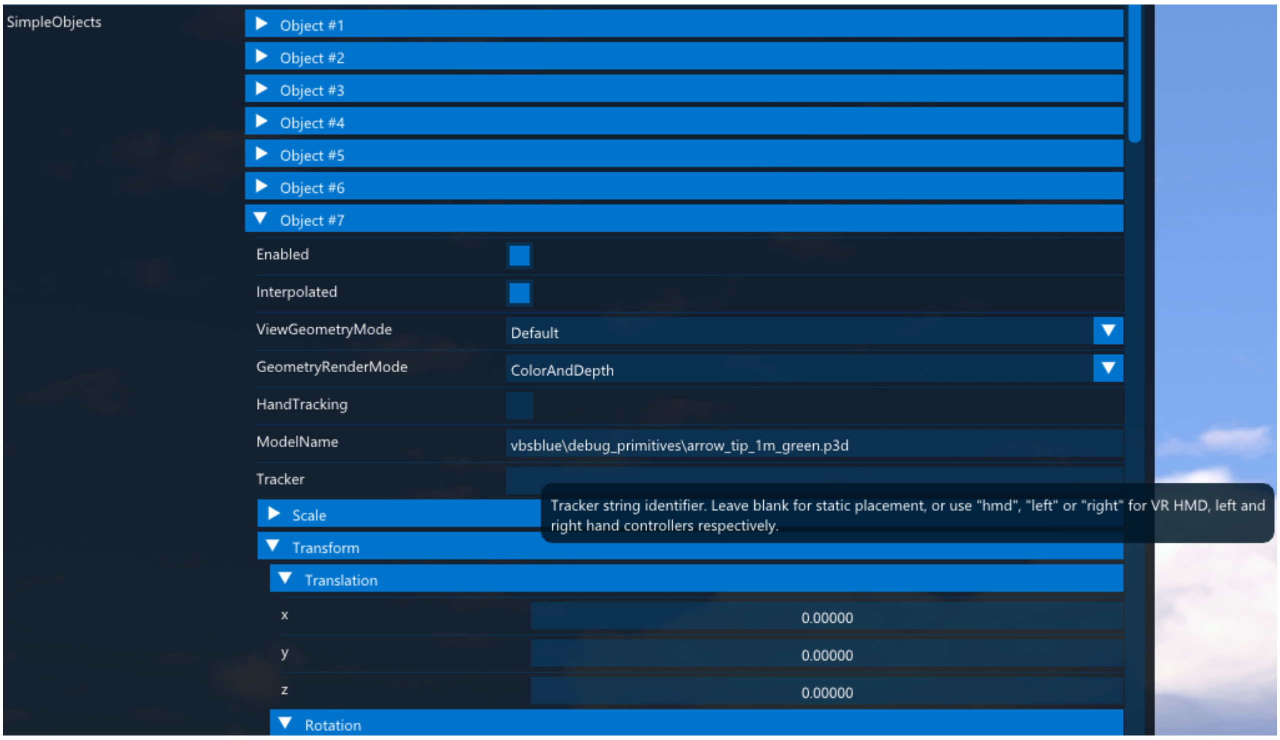
4. Select **OpenVR Trackers** from the Debug UI.

5. If the tracker is currently active and connected, it should be visible on this dialog.

> **NOTE**
> If the tracker is not visible, ensure that it is turned on and paired to SteamVR.

6. Find the tracker, expand it and click **Copy Tracker Name to Clipboard**.
7. Press **Tab** to open the Settings UI.
8. Select **XR** > **TrackingSpace** > **Transform** to show the saved and stored calibration.

9. The tracker name can be pasted in the tracker name section within the XR setting Object **Tracker** field.



10. Click **Save Settings** in the Settings dialog.

> **NOTE**
> Leaving the Tracker field blank attaches this object to the parent platform, at either the platform object's origin or offset (if specified).

11. **Optional**: Consider the following alternatives to using the tracker name.
    - Adding the keyword *hmd* in the Tracker field attaches the given object to the HMD.

      > **NOTE**
      >
      > Objects placed directly in the line of sight of HMD users may **float** due to a fixed frame latency when synchronizing model spatials with a separate double-buffered rendering thread. It also does not account for SteamVR reprojection (timewarp).

    - Adding **left** or **right** in the Tracker field attaches the object to the HMD, left controller or right controller, if available.

      > **NOTE**
      >
      > Attaching objects to the HMD may float since VBS Blue IG model spatial processing is 1 frame behind the HMD, and does not account for any SteamVR reprojection.

    - TrackerNet support
      - If **VR / Networking / EnableRecv** is enabled, any remote tracker paired to another IG / VBS4 client on the same network using the same TrackerNet Multicast address can be entered here. The remote client(s) must have **EnableSend** enabled before they can broadcast their own tracker positions.
      - In addition to the unique tracker device names, special keywords are also supported to handle specific remote devices:

        a. **HMD_[client]**: Other client HMD transforms. **Example**: HMD transform for client #2 is **HMD_2**.

        b. **LController_[client]**: Other client left hand controller.

        c. **RController_[client]**: Other client right hand controller.

        > **NOTE**
        >
        > Connected TrackerNet client IDs must be unique from one another.

4.5.8 Tracked Object Types

Tracked objects refer to virtual objects in platform space that can either track the movement of hardware tracking devices in tracking space, or be fixed at some arbitrary location in platform space. These objects have additional fixed transformation offsets, scale, and even attachments. Tracked objects contain unique Tracked Object Setup (on page 118) , and include the following object types:

- Simple Objects (below)
- Weapons (on the next page)
- Checklists (on the next page)
- Masks (on page 117)

4.5.8.1 Simple Objects

Simple objects correspond to an arbitrary model. This can either be a `p3d` model path or a `CfgVehicle` class type. These can either be attached to a tracker or fixed to the platform, both with configurable transform offset and scaling.

4.5.8.2 Weapons

Weapons extend the functionality of tracked objects by sending trigger events with calculated muzzle transforms based on the tracker position. Additionally, weapons can support attachments including optics, lasers, and mixed-reality masks.

4.5.8.3 Checklists

Checklists are special surfaces containing HTML. They are primarily intended as checklists but can be used for other signage.

4.5.8.4 Masks

Masking meshes are mixed-reality volumes that can be used to mask in / out real-world objects (via camera or some other method) to composite real imagery against the virtual environment. Usage requires mixed-reality capable headsets, such as the Varjo XR-3.

For information on configuring these objects, see Tracked Object Setup (on the next page).

4.5.9 Tracked Object Setup

Configure different tracked object types according to the procedures below:

- Simple Object (below)
- Weapon Setup (on page 120)
- Checklist Setup (on page 129)
- Masking Mesh Setup (on page 130)

4.5.9.1 Simple Object

Simple objects in XRTP have little complexity. The model created (either a `P3D` path or `CfgVehicles` class) attaches directly to the tracker origin. Any transformation offsets or scaling applied is performed within the local coordinate system for the tracker.

Simple objects are configured under the SimpleObjects array in the XR settings.

Setting Name	Description
Enabled	Toggles visibility of the model.
Interpolated	If enabled, this option smoothes tracking jitter at the expense of increased lag.
ViewGeometryMode	Visual LOD for the model. Usually **pilot** has the highest detail. Intended for advanced users.
GeometryRenderMode	Alternative rendering methods to render the model depth-only or color-only for mixed-reality compositing. Intended for advanced users.
ModelName	Content p3d model (including path) or CfgVehicles classname.
Tracker	Name of the tracker. See Tracker Name Field for more information on how to assign trackers.
Scale	Scaling along the object's axes.
Transform / Translation	Object offset translation, in meters.
Transform / Rotation	Model space rotation offset angles (yaw, pitch, roll) in degrees.

Add a simple object to the tracking space in VBS Blue IG using the procedure below.

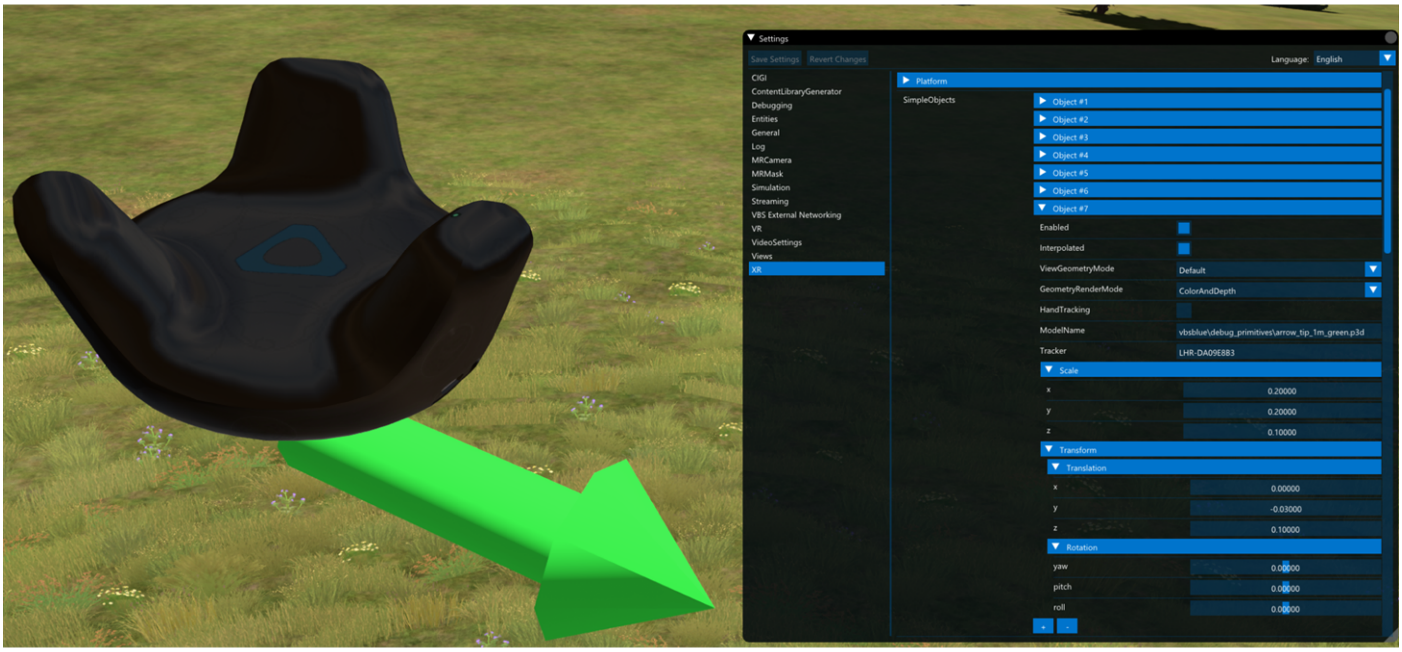
Follow these steps:

1. Press **Tab** to open the Settings UI.
2. Select **XR** > **SimpleObjects** list.
3. In **SimpleObjects**, click **+** to create a new Object entry.
4. Click the Object dropdown.
5. In the **ModelName** field, type the `CfgVehicles` class name or `P3D` filename (with prefix / path) to render the model.

   For this example, use:

   `vbsblue\debug_primitives\arrow_tip_1m_green.p3d`
6. In the **Tracker** field, enter the name.

   > **NOTE**
   >
   > See Tracker Name Field for more information on this field, as this is unique to the hardware device you're currently using.
7. Adjust **Translation** and **Rotation** within the parent coordinate space for the object, whether it is a tracker or the platform offset.



4.5.9.2 Weapon Setup

Compared to simple objects, weapons have more complexity due to slot instancing as well as spatial hierarchy. Configure weapon object types using the following topics:

- Weapon Spatial Hierarchy (on the next page)
- Weapon Slots (on the next page)
- Weapon Attachments (on page 123)
- Configuring Weapon Objects (on page 124)
- Example Weapon Configuration - M4 (on page 126)

4.5.9.2.1 Weapon Spatial Hierarchy

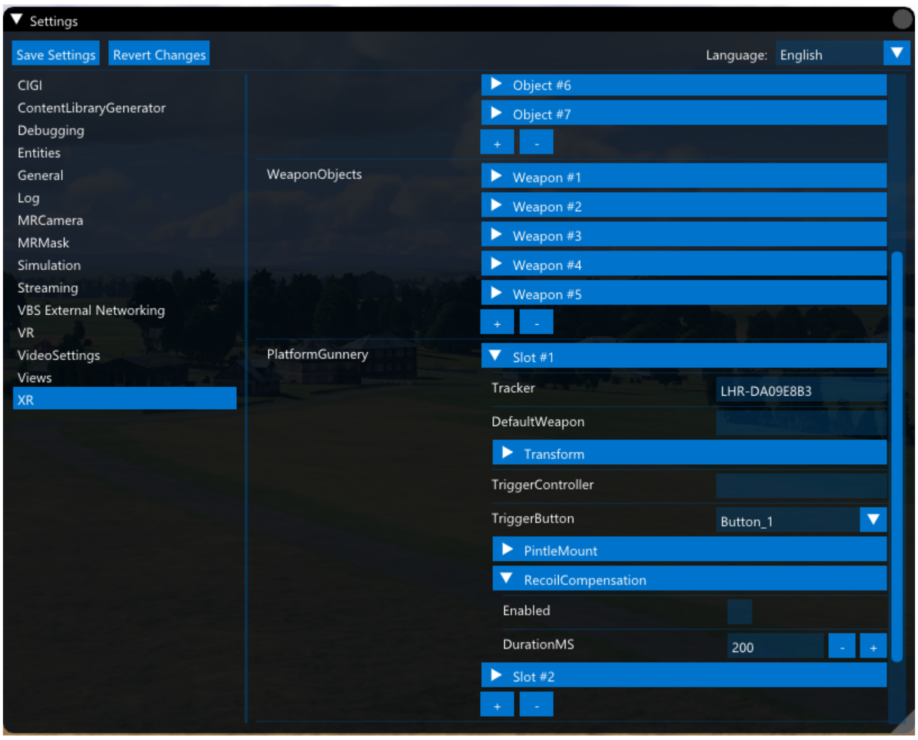
- Tracker (root)
    - Gunnery Transform
        - WeaponObject (model / alias) Transform
            - Muzzle Transform

              Likely to be only used as fallback in the future and rely on *muzzle* memory point for CfgWeapon objects.
            - Attachment(s) Transform
                - Optic Symbology Transform
                - Mask Transform
            - Laser Transform

4.5.9.2.2 Weapon Slots

Weapons can be instanced so that multiple weapons of the same type can be attached to multiple pucks.

Each slot is determined under PlatformGunnery in the XR settings. Slots correspond to physical tracker and trigger control (if applicable) for each weapon.



Setting Name	Description
Tracker	Name of the tracker. See Tracker Name Field for more information on how to assign trackers.
DefaultWeapon	Assigned weapon on initial platform attached. Leave blank to have no weapon assigned at startup. Assigning this at runtime requires an interop host component function, XR Training Platform debug UI, or special SDK function.
Transform	Slot-specific transforms applied in addition to the weapon type-specific offsets.
TriggerController	Controller Input UUID for the trigger device, if applicable. Leave blank for no trigger support.
TriggerButton	Button ID for the trigger, only applicable if TriggerController was set.
PintleMount	Intended for mounted turret weapons, supports adding a model and restricting movement of the tracked weapon so that it remains locked to a virtual pivot point.
RecoilCompensation	Additional settings to compensate for IMU error experienced during high G-forces. Specifically, some mechanical recoil systems can momentarily exceed the 8G or 16Gb limits of certain SteamVR trackers, leading to severe drifting or other errors.

4.5.9.2.3 Weapon Attachments

Weapons support attachment models with configurable offsets.

Setting Name	Description
Enabled	Toggles the default visibility of the attachment when the weapon is assigned to the slot. If unchecked, the attachment model, mask and any associated OpticSymbols are also hidden
ModelName	P3D model path of the attachment model. Can be left blank if nothing needs to be rendered directly, and only consists of a mask and / or OpticSymbol(s).
Transformation	Fixed translation and rotation offset of this attachment relative to its parent weapon.
Mask / Enabled	Toggles the visibility of a mixed-reality masking mesh for this weapon attachment. (Requires Mixed-Reality cmdline and supported hardware).
Mask / id	Masking mesh ID, if enabled. **NOTE** A conflict can occur if this ID is the same as another attachment or MaskingMesh ID. At this time, instancing support for masking meshes is unavailable.
Mask / Transform	Translation and rotation of the masking mesh relative to the parent attachment model.
Mask / Scale	Scale of the masking mesh attachment.

4.5.9.2.4 Configuring Weapon Objects

Setting Name	Description
Interpolated	If checked, this enables motion smoothing of the visual weapon object.
ModelName	CfgWeapons class name or p3d path model for the weapon. If using a class name, this string is used when triggering the weapon to the interop host, and used to determine the default munition type and other simulated behaviors.
Visuals / ViewGeometryMode	Visual LOD for the weapon. Usually **pilot** is the highest-detailed setting. Intended for advanced users.
Visuals / GeometryRenderMode	Alternative rendering methods to render the weapon model depth-only or color-only for mixed-reality compositing. Intended for advanced users.
Visuals / HideChildren (array)	Array of strings corresponding to any child proxies (components) to be removed. For example, having **vbs_xx_m4_carryhandle_blk** for a **vbs_xx_m4a1_blk** weapon spawns the weapon with the carry handle removed.
Visuals / Alias / Enabled	Enables a model alias. If checked, this changes the appearance of the weapon on the client. However, any interop host simulated behavior and gunnery slot lookup remains based on Weapon / ModelName.
Visuals / Alias / ModelName	If Visuals / Alias/Enabled is checked, this determines which visual model is rendered instead of Weapon / ModelName. If left blank, the weapon model is hidden (any enabled attachment still renders, however).

Setting Name	Description
Scale	Scales the weapon model. Does not affect attachments, laser or muzzle transforms.
Transform	Translation and Rotation of the weapon model. Affects attachments, laser, and muzzle transforms.
MuzzleOffset	Positional offset of the weapon's muzzle. Only applicable for the interop host simulation, as this is where the simulated munition originates when the weapon is fired.
TriggerHold / Enabled	Enables TriggerHold events so that weapon muzzle transform and trigger state is continuously sent while the mechanical trigger is depressed. To reduce unnecessary weapon spamming for non-automatic weapons, this should only be enabled for burst or full-auto capable weapons. **NOTE** This is not intended to toggle the semi / burst / full auto mode of the weapon.
TriggerHold / PollingMS	If the TriggerHold events are enabled, this option throttles network events so that the event is sent no less than N milliseconds. For example, 33.333 ms (default) means that this event is sent to the host a maximum of 30 times / second.
Laser (group)	Group of weapon-specific laser attachment settings to modify color, beam settings, and offset from the weapon.
Attachment (array - group)	An arbitrary array of attachments with visual models and offsets. Usually indicated to have optics with them.
Attachment / OpticsSymbology	An array of groups to add optical symbology, such as crosshairs, red dots, or other features. Can be fixed or collimated.

4.5.9.2.5 Example Weapon Configuration - M4

The example below is the M4 carbine. However, most kinds of supported weapon class (under CfgWeapons) can be used, including rifles, handguns, heavy weapons, and even some mounted guns.

Follow these steps:

1. Select a suitable location to mount the tracker.
    - Tracker can be mounted in any orientation or offset; this tracker-weapon offset is accounted for in later steps.
    - The method of attaching the tracker to the weapon training device is left to the user. However, it is advisable to ensure that the tracker mount is sturdy, and preferably made of strong, nonmagnetic material, such as aluminum.
    - Many weapons usually have standard mounting support for optics or other attachments. All of the supported trackers come with a standard ¼-inch camera screw base.

2. Press **Tab** to open the VBS Blue IG Settings (on page 201).
3. Select **XR**.
4. In the **WeaponObjects** option, click **+** to create an entry .
5. Enter **vbs_xx_m4a1_blk** under **ModelName**.
6. Change **ViewGeometryMode** to **Pilot**.
7. Add an entry to **HideChildren** with the name **vbs_xx_m4_carryhandle_blk**.
8. Based on where the puck is located on the weapon, adjust the Translation and Rotation under the weapon **Transform**.

   In our specific example (tracker mounted on right-front of rail), the following values are used:

   i. **Translation**: (0.077, -0.05, -0.18) for 7.7 cm left, 5 cm down, and 18 cm back.

   ii. **Rotation**: (0, 0, -90) for rolling the model 90 degrees counterclockwise from the tracker.
9. Scroll down to **PlatformGunnery**. If the list is empty, add a new **WeaponSlot** entry using the **+** button.
10. Under this new **Slot**, add the tracker name under the **Tracker** field.
11. Click on **Save Settings**.
12. To make the XRTP debug UI functionality available, click **Enable** at the top of XR Settings UI.
13. Press backquote (`) / tilde (**~**) on the keyboard to open the **Debug UI**.
14. Click the **XR Training Platform** option to open the debug window.
15. If an ownship has not been created yet, click **Create Debug Ownship**.
16. Under the active ownship, select the weapon `vbs_xx_m4a1_blk` from the weapon combo box for **Gun Slot #1** and click **Assign Weapon Slot** button.

    The M4 rifle should be visible now and attached to the tracker.
17. On the Debug UI, expand the **Weapons** section, expand the weapon slot and enable **Show Debug Visuals**T.

    A visual aid appears. This is for the muzzle offset.

18. Adjust the muzzle offset in the Weapon settings so that it corresponds to the muzzle location on the model.

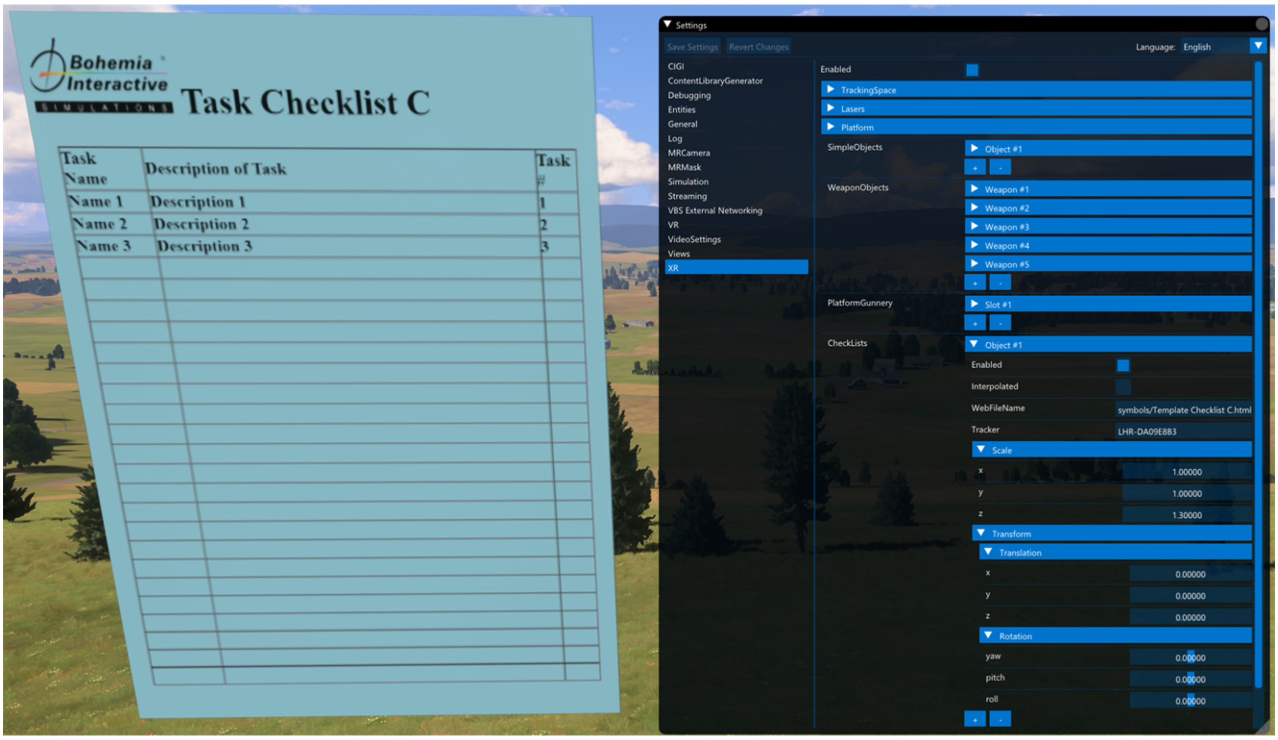
    

    For more setup options, see the XML file at:

    `\IG_Installation\Component\XRTrainingPlatform\ExampleSettings\XR.xml`.

4.5.9.3 Checklist Setup

Checklists operate similarly to a Simple Object type, except checklists render a 2D HTML page instead of a model.



The procedure below creates a checklist object.

Follow these steps:

1. Navigate to the **Checklists** array in the XR Settings UI.
2. Click **+** to add a new entry.
3. A default webpage shows up by default.
   - This webpage is locally stored in the following directory:

     `\IG_Installation\Components\XRTrainingPlatform\symbols\`
   - Currently, input events are not supported.
   - The page is rendered with the backface, and appears reversed.

     To render 2-sided, a second checklist can be duplicated with yaw flipped 180 degrees and a Z translation of 0.001 to prevent Z-fighting.

4.5.9.3.1 Checklist Settings

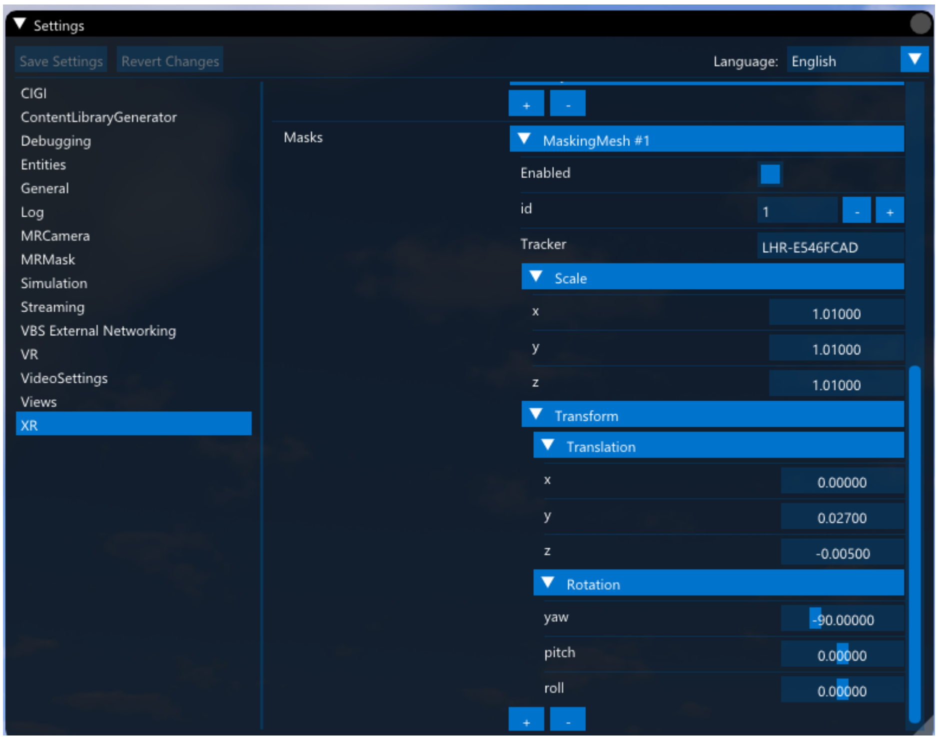
Setting Name	Description
Enabled	Toggles the visibility of the checklist object.
Interpolated	If checked, this enables motion smoothing of the visual checklist.
WebFileName	HTML file path relative to: `\IG_Installation\Components\XRTrainingPlatform\`
Tracker	Name of the tracker that the checklist is attached to. See Tracker Name Field for more information on how to assign trackers.
Scale	X, Y scale the width and height of the quad. Z is the shrinking within the web page itself. For examples, 1 = 100%, 1.5 = 75%, 2 = 50% scale.
Transform	Translation and Rotation of the checklist respective to the tracker / parent.

4.5.9.4 Masking Mesh Setup

Masking meshes create 3D geometry for mixed-reality masking. This requires the `-mask` hmd parameter, with the mask meshes loaded from:

`\IG_Installation\Systems\HMD\Assets\`

Only one mask of a given ID can be instanced at a time; two masking meshes sharing the same ID cause conflicts.



Setting Name	Description
Enabled	Toggles the visibility of the mask.
id	Masking mesh ID. IDs are enumerated by the filename number.
Tracker	Name of the tracker that the mask is attached to. See Tracker Name (on page 111) for more information on how to assign trackers.
Scale	Scale of the masking mesh.
Transform	Translation and Rotation of the masking mesh relative to the tracker.

4.6 Warping on Curved Displays

VBS Blue IG allows you to warp and blend image output from multiple VBS Blue IG views to display images on large curved surfaces, such as domes.

The following methods are supported:

- **Scalable Warping** - Warp and Blend integration with Scalable Display Manager.
- **DomeProjection Warping** - Warp and Blend integration with DomeProjection ProjectionTools.
- **Generic Warping** - Generic view overrides, for integration with external warp and blend solutions (for example, ImmersaView SimVisuals).

Each method requires the configuration to specify the views to be modified. Configuration is enabled by the combination of a Startup Parameters (on page 454) and modification of the Views and Render Targets (on page 183).

See the following topics for configuration of each warping method:

- Scalable Warping Setup (on page 140) - Describes how to set up Scalable Display Manager with VBS Blue IG.
- DomeProjection Warping Setup (on page 137) - Describes how to set up DomeProjection ProjectionTools with VBS Blue IG.
- Generic Warping Setup (on the next page) - Describes how to set up a generic warping solution with VBS Blue IG.

Once the warping is configured, see the related VBS Blue IGUI controls:

- Warping Status (on page 427) - Describes the Warping Status panel, which is part of the VBS Blue IG Debug UI (on page 376) and contains UI controls for each warping solution.

4.6.1 Generic Warping Setup

Generic Warping allows for overriding a VBS Blue IG view frustum, position offset, and orientation offset, to provide a simplified method of integration with an external warp and blend solution. It does not perform any warping or blending itself.

This can allow for integration with warping and blending software that operates on the desktop, or otherwise injects with VBS Blue IG. For example, ImmersaView SimVisuals.

The following approaches are supported:

- **Single view per VBS Blue IG instance.**
  - VBS Blue IG is run on a PC, with a single render target and viewport.
  - Multiple PCs are then used to provide multiple views.

> **NOTE**
> This is the recommended approach for performance.

- **Multiple viewports per VBS Blue IG instance.**
  - VBS Blue IG is run on a PC with a single window render target containing multiple viewports. The render target is stretched to span across multiple displays.
  - Multiple PCs can be used for additional views.

> **NOTE**
> This option is available if limited hardware is available. Performance will be impacted.

- **Multiple render targets per VBS Blue IG instance.**
  - This option is available if limited hardware is available. Performance will be impacted.
  - VBS Blue IG is run on a PC, with multiple window render targets, each containing one or more viewport. Each window render target is positioned and sized to fill each display.

As mentioned, this only provides a method to override a view's frustum, position offset, and orientation offset. If the simulation host already specifies the correct view information, this method is not required.

Follow these steps:

1. Modify `DefaultViewConfig.xml` for each VBS Blue IG instance.
   a. Create each view with a GenericWarp XML node. For sample code , see Using single view per VBS Blue IG instance (on the next page).
   b. If necessary, the view can be configured to target a specific warp definition file, which is required when using multiple views per VBS Blue IG instance.

      By default, the warp definition file is in the following folder:

      `\IG_Installation\Components\BlueIGViewSystems\GenericWarp\GenericWarpDefinition\`

      For sample code , see Using multiple viewports per VBS Blue IG instance (on the next page).
   c. Alternatively, use multiple window render targets to show a single viewport. Each window render target is then configured to display on separate display projectors.

      For sample code , see Using multiple render targets per VBS Blue IG instance (on page 136).
2. Configure each generic warp definition file with the necessary overrides. A sample generic warp definition file is provided in:

   `\IG_Installation\Components\BlueIGViewSystems\GenericWarp\/GenericWarpDefinition\`

   a. The Field Of View is a direct override of the view's field of view.
   b. The Position and Orientation are offsets to the view's position and orientation.
   c. The method to apply the orientation offset can additionally be configured (for example, Yaw-Pitch-Roll vs Roll-Pitch-Yaw).
3. Launch VBS Blue IG with the `-warp` command-line parameter (see Startup Parameters (on page 454)) for eachVBS Blue IG client.
4. When the simulation host creates the specified views, the views and any additionally configured window render targets will be created, and the views will be overridden as configured by default.
5. If necessary, the external warp and blend software should be configured as required.

4.6.1.1 Code Snippets

- Using single view per VBS Blue IG instance (below)
- Using multiple viewports per VBS Blue IG instance (below)
- Using multiple viewports per VBS Blue IG instance (below)

Using single view per VBS Blue IG instance

Each view should contain the GenericWarp XML node. Use the following example to enable View ID #2:

```
<Config>
  <Views>
    <View ID="2">
      <RenderMethod Method="Screen" />
      <GenericWarp Enabled="True" Definition="GenericWarp\GenericWarpDefinition.xml"
/>
    </View>
  </Views>
</Config>
```

Using multiple viewports per VBS Blue IG instance

The following example demonstrates a case where 3 display projectors are being used. 3 views are configured to display on a single window render target, with each view's viewport configured to use a third of main window render target. The main window render target would then be stretched to span the entire desktop output, so each display projector would show a single viewport.

```
<Config>
  <Views>
    <View ID="0">
      <RenderMethod Method="Screen" />
      <Viewport Left="0.0" Top="0.0" Right="0.333333" Bottom="1.0" Layer="0" />
      <GenericWarp Enabled="True" Definition="GenericWarp\View0.xml" />
    </View>
    <View ID="1">
      <RenderMethod Method="Screen" />
      <Viewport Left="0.333333" Top="0.0" Right="0.666666" Bottom="1.0" Layer="0" />
      <GenericWarp Enabled="True" Definition="GenericWarp\View1.xml" />
    </View>
    <View ID="2">
      <RenderMethod Method="Screen" />
      <Viewport Left="0.666666" Top="0.0" Right="1.0" Bottom="1.0" Layer="0" />
      <GenericWarp Enabled="True" Definition="GenericWarp\View2.xml" />
    </View>
```



```
  </Views>
</Config>
```

Using multiple render targets per VBS Blue IG instance

The following example shows an alternative approach with three window render targets used instead of a single large window render target spanning the entire desktop.

```
<Config>
  <Views>
    <View ID="0">
      <RenderMethod Method="Screen" />
      <RenderTarget ID="0" />
      <GenericWarp Enabled="True" Definition="GenericWarp\View0.xml" />
    </View>
    <View ID="1">
      <RenderMethod Method="Screen" />
      <RenderTarget ID="1" />
      <GenericWarp Enabled="True" Definition="GenericWarp\View1.xml" />
    </View>
    <View ID="2">
      <RenderMethod Method="Screen" />
      <RenderTarget ID="2" />
      <GenericWarp Enabled="True" Definition="GenericWarp\View2.xml" />
    </View>
  </Views>
  <RenderTargets>
    <RenderTarget Type="Window" ID="1">
      <Window X="-1920" Y="0" WindowWidth="1920"
              WindowHeight="1080" IsBorderless="True" />
    </RenderTarget>
    <RenderTarget Type="Window" ID="2">
      <Window X="1920" Y="0" WindowWidth="1920"
              WindowHeight="1080" IsBorderless="True" />
    </RenderTarget>
  </RenderTargets>
</Config>
```

Generic Warping is configured and VBS Blue IG is ready to project an image to a dome, and you can now use the Warping Status (on page 427) UI controls in VBS Blue IG.

4.6.2 DomeProjection Warping Setup

The DomeProjection integration allows VBS Blue IG to blend and warp image output from multiple VBS Blue IG views to display images on large curved surfaces such as domes. Each VBS Blue IG view provides a visual channel representing a segment of the full image, which combines to form one seamless image when properly configured.

The following approaches are supported:

- Single view per VBS Blue IG instance.
  - VBS Blue IG is run on a PC, with a single render target and viewport.
  - Multiple PCs are then used to provide multiple views.

  > **NOTE**
  > This is the recommended approach for performance.

- Multiple viewports per VBS Blue IG instance.
  - VBS Blue IG is run on a PC with a single window render target containing multiple viewports. The render target is stretched to span across multiple displays.
  - Multiple PCs can be used for additional views.

  > **NOTE**
  > This option is available if limited hardware is available. Performance will be impacted.

Use DomeProjection ProjectionTools to configure each display channel. See the documentation with ProjectionTools for more information.

DomeProjection ProjectionTools will create configuration in `C:\DomeProjection\data\config.xml` on each computer. The channel IDs listed in this file are as configured in ProjectionTools, and will be used to link a channel configuration to a view in VBS Blue IG.

Follow these steps:

1. Confirm that the DomeProjection configuration process created `config.xml` files in `C:\DomeProjection\data` on each computer.

2. Modify `DefaultViewConfig.xml` for each VBS Blue IG instance.
   a. Each view for enabling DomeProjection Warp and Blend should contain the DomeProjection XML node. In the following example, configure View ID #2 to use the configuration for channel #2:

```
<Config>
  <Views>
    <View ID="2">
      <RenderMethod Method="Screen" />
      <DomeProjectionWarp Enabled="True" ChannelID="2" />
    </View>
  </Views>
</Config>
```

   b. The following example demonstrates a case where three display projectors are used. Three views are configured to display on a single window render target, with the viewport of each view configured to use a third of main window render target. The main window render target is then stretched to span the entire desktop output, so each display projector shows a single viewport.

```
<Config>
  <Views>
    <View ID="0">
      <RenderMethod Method="Screen" />
      <Viewport Left="0.0" Top="0.0" Right="0.3333" Bottom="1.0" Layer="0"
/>
      <DomeProjectionWarp Enabled="True" ChannelID="0" />
    </View>
    <View ID="1">
      <RenderMethod Method="Screen" />
      <Viewport Left="0.3333" Top="0.0" Right="0.6667" Bottom="1.0"
Layer="0" />
      <DomeProjectionWarp Enabled="True" ChannelID="1" />
    </View>
    <View ID="2">
      <RenderMethod Method="Screen" />
      <Viewport Left="0.6667" Top="0.0" Right="1.0" Bottom="1.0" Layer="0"
/>
      <DomeProjectionWarp Enabled="True" ChannelID="2" />
    </View>
  </Views>
</Config>
```

3. Launch VBS Blue IG with the `-domeprojection` command-line parameter (see Startup Parameters (on page 454)) for each VBS Blue IG client.

4. When the simulation host creates the specified views, the views and any additionally configured window render targets are created, and the views are warped and blended as configured by default.

DomeProjection Warping is now configured and VBS Blue IG is ready to project an image to a dome. Optionally consider using the Warping Status (on page 427) debugging controls in VBS Blue IG.

4.6.3 Scalable Warping Setup

The Scalable integration allows VBS Blue IG to blend and warp image output from multiple VBS Blue IG views to display images on large curved surfaces such as domes. Each VBS Blue IG view provides a visual channel representing a segment of the full image, all of which combine to form one seamless image, when properly configured.

The following approaches are supported:

- **Single view per VBS Blue IG instance.**
  - VBS Blue IG is run on a PC, with a single render target and viewport.
  - Multiple PCs are then used to provide multiple views.

> **NOTE**
> This is the recommended approach for performance.

- **Multiple viewports per VBS Blue IG instance.**
  - VBS Blue IG is run on a PC with a single window render target containing multiple viewports. The render target is stretched to span across multiple displays.
  - Multiple PCs can be used for additional views.

> **NOTE**
> This option is available if limited hardware is available. Performance will be impacted.

- **Multiple render targets per VBS Blue IG instance.**
  - This option is available if limited hardware is available. Performance will be impacted.
  - VBS Blue IG is run on a PC, with multiple window render targets, each containing one or more viewport. Each window render target is positioned and sized to fill each display.

Use Scalable Display Manager to configure each display channel. See the documentation with Scalable Display Manager for more information.

> **NOTE**
> Ensure that Black Level, Masking and Color are disabled in Scalable Display Manager. These features are not currently supported with VBS Blue IG.

Scalable Display Manager creates a configuration on each computer at:

`C:\Program File\Scalable Display\DEI\LocalCalibration\`

Follow these steps:

1. Confirm that the Scalable configuration process created `ScalableData.ol` files on each computer at:

   `C:\Program File\Scalable Display\DEI\LocalCalibration\`

2. Modify `DefaultViewConfig.xml` for each VBS Blue IG instance.

   a. To enable Scalable Warp and Blend, create each view with the ScalableWarp XML node.

      For sample code , see Using single view per VBS Blue IG instance (below).

   b. If necessary, the view can be configured to target a specific mesh file. This is required when using multiple views per VBS Blue IG instance.

      By default, the Scalable mesh file is the following:

      `C:\Program File\Scalable Display\DEI\LocalCalibration\ScalableData.ol`

      For sample code , see Using multiple viewports per VBS Blue IG instance (on the next page).

   c. Alternatively, multiple window render targets could be used to show a single viewport. Each window render target is then configured to display on each display projector.

      For sample code , see Using multiple render targets per VBS Blue IG instance (on the next page).

3. Launch VBS Blue IG with the `-scalable` command-line parameter (see Startup Parameters (on page 454)) for each VBS Blue IG client.

4. When the simulation host creates the specified views, the views and any additionally configured window render targets will be created, and the views will be warped and blended as configured by default.

4.6.3.1 Code Snippets

- Using single view per VBS Blue IG instance (below)
- Using multiple viewports per VBS Blue IG instance (on the next page)
- Using multiple viewports per VBS Blue IG instance (on the next page)

Using single view per VBS Blue IG instance

Each view to enable Scalable Warp and Blend on should contain the ScalableWarp XML node. Use the following example to enable View ID #2:

```
<Config>
  <Views>
    <View ID="2">
      <RenderMethod Method="Screen" />
```



```
      <ScalableWarp Enabled="True" />
    </View>
  </Views>
</Config>
```

Using multiple viewports per VBS Blue IG instance

The following example demonstrates a case where 3 display projectors are being used. 3 views are configured to display on a single window render target, with each the viewport of each view configured to use a third of main window render target. The main window render target would then be stretched to span the entire desktop output, so each display projector would show a single viewport.

```
<Config>
  <Views>
    <View ID="0">
      <RenderMethod Method="Screen" />
      <Viewport Left="0.0" Top="0.0" Right="0.333333" Bottom="1.0" Layer="0" />
      <ScalableWarp Enabled="True" MeshFile="C:\Program Files\Scalable
Display\DEI\LocalCalibration\ScalableData.ol" />
    </View>
    <View ID="1">
      <RenderMethod Method="Screen" />
      <Viewport Left="0.333333" Top="0.0" Right="0.666666" Bottom="1.0" Layer="0"
/>
      <ScalableWarp Enabled="True" MeshFile="C:\Program Files\Scalable
Display\DEI\LocalCalibration\ScalableData.ol_1" />
    </View>
    <View ID="2">
      <RenderMethod Method="Screen" />
      <Viewport Left="0.666666" Top="0.0" Right="1.0" Bottom="1.0" Layer="0" />
      <ScalableWarp Enabled="True" MeshFile="C:\Program Files\Scalable
Display\DEI\LocalCalibration\ScalableData.ol_2" />
    </View>
  </Views>
</Config>
```

Using multiple render targets per VBS Blue IG instance

The following example shows an alternative approach with three window render targets used instead of a single large window render target spanning the entire desktop.

```
<Config>
  <Views>
    <View ID="0">
      <RenderMethod Method="Screen" />
      <RenderTarget ID="0" />
```



```
      <ScalableWarp Enabled="True" MeshFile=
        "C:\Program Files\Scalable Display\DEI\LocalCalibration\ScalableData.ol" />
    </View>
    <View ID="1">
      <RenderMethod Method="Screen" />
      <RenderTarget ID="1" />
      <ScalableWarp Enabled="True" MeshFile=
      "C:\Program Files\Scalable Display\DEI\LocalCalibration\ScalableData.ol_1" />
    </View>
    <View ID="2">
      <RenderMethod Method="Screen" />
      <RenderTarget ID="2" />
      <ScalableWarp Enabled="True" MeshFile=
      "C:\Program Files\Scalable Display\DEI\LocalCalibration\ScalableData.ol_2" />
    </View>
  </Views>
  <RenderTargets>
    <RenderTarget Type="Window" ID="1">
      <Window X="-1920" Y="0" WindowWidth="1920"
              WindowHeight="1080" IsBorderless="True" />
    </RenderTarget>
    <RenderTarget Type="Window" ID="2">
      <Window X="1920" Y="0" WindowWidth="1920"
              WindowHeight="1080" IsBorderless="True" />
    </RenderTarget>
  </RenderTargets>
</Config>
```

Scalable Warping is configured and VBS Blue IG is ready to project an image to a dome, and you can now use the Scalable Warping (on page 427) UI controls in VBS Blue IG.

5. Systems

This section provides details about various systems contained within VBS Blue IG, their corresponding configuration and methods of control.

- Audio Engine (on page 379) - Set up a scenario to play back and add sound effects.
- Automatic Animations (on page 149) - Automatically animate features on Lifeform and Object and Vehicle entities.
- Symbology (on page 159) - Create a symbology configuration using standard and custom CIGI packets, and Lua scripting.
- Mirrors and Periscopes (on page 151) - Set up mirror and periscope viewing for relevant vehicle models.
- Ropes (on page 157) - Create and control simulation and rendering of rope objects via CIGI packets.
- Thermal Simulation Control (on page 168) - Control thermal values directly through a host simulation or have them automatically calculated within VBS Blue IG.
- Time and Time Progression (on page 169) - Control several independent time systems.
- Tracks and Trails (on page 171) - Configure Vehicle Tracks, Dust Trails, and Lifeform Tracks.
- Views and Render Targets (on page 183) - Configure multiple views and targets in a single instance.

5.1 Audio Engine

VBS Blue IG contains basic support for playback of sound effects. The application allows for playing back and adding simple stereo sounds, or more advanced positional 3D audio with up to 7.1 channel support.

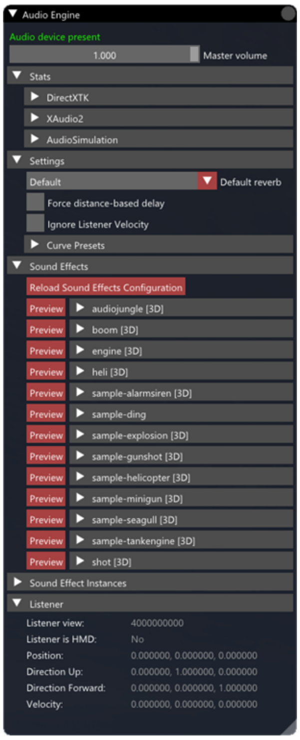
Use these procedures for making use of the Audio Engine:

- Debug UI Testing (below)
- Adding Sounds (on the next page)
- CIGI Playback (on page 147)

5.1.1 Debug UI Testing

The Audio Engine window within the Debug UI provides the following options:

- Monitoring of audio usage stats.
- Modify and test runtime audio settings such as reverb, delay, and velocity.
- Preview sound effects.



Follow these steps:

1. Press backquote ` / tilde ~ to access the Debug UI (on page 376).
2. Select **Audio Engine**.

3. Expand any of the following:

Parameters Section	Description
Master Volume	Adjust sound level of master volume setting.
Stats	Select any of the following to view statistical data for the item: • **DirectXTK** • **XAudio2** • **AudioSimulation**
Settings	• **Default Reverb** - Select a Default Reverb option type from the drop-down list. • **Force distance-based delay** - Click to enable / disable. • **Ignore Listener Velocity** - Click to enable / disable.
Sound Effects	• **Reload Sound Effects Configuration** - Reloads the configuration of all sound effects. **NOTE** Pre-existing sound effects will not be effected by reloading, only new effect instances added. To add new files, see Adding Sounds (below). • **Preview** - Preview the sound effect by playing it as a one-shot sound.
Sound Effect Instances	Displays any instance of sound effects in use.
Listener	Displays information about Listeners added to an IG instance.

5.1.2 Adding Sounds

Follow these steps:

1. Add a .wav file containing your sound to the following directory:

   `\IG_Installation\Components\AudioEngine\Sounds\`

   If the Sounds directory does not exist, create it.

2. Add a `MySounds.xml` file (the name of the XML file does not matter) containing a group of mappings between the name of the sound and the .wav file, as well as additional properties for the sound effects. A basic sample containing the available parameters is listed below.

```
<?xml version="1.0"?>
<SoundEffects>
  <SoundEffect3D name="heli" file="heli.wav" looping="true" volume="1.0"
pitch="0.0"
   innerRadius="0.0" innerRadiusAngle="0.0" curveDistanceScaler="5000"
dopplerScaler="1.0" />
  <SoundEffect3D name="boom" file="boom.wav" curveDistanceScaler="5000.0" />
  <SoundEffect3D name="shot" file="shot.wav" curveDistanceScaler="1200.0" />
  <SoundEffect3D name="engine" file="engine.wav" curveDistanceScaler="1200.0"
looping="true" volume="1.0" pitch="0.0" pan="0.0" />
  <SoundEffect name="mouseclick" file="click.wav" />
</SoundEffects>
```

5.1.3 CIGI Playback

Sound effects can be controlled by a CIGI host. Sounds are created with the **Entity Control** packet, using a **Sound** entity type. The **Animation State** parameter (in the **Animation Control** packet in CIGI 4.0) controls playback.

Follow these steps:

1. Open the Cigi Protocol settings at:

   `%LOCALAPPDATA%\Bohemia Interactive Simulations\VBS Blue IG\<version>\Settings\CIGI.xml`.

2. Modify the Queries setting to `<Queries Enabled="false">`.
3. Add a new entity mapping in the file, using the sound identifier `<sound name>`.

```
cigi-entity-type:50000 > sound:heli
cigi-entity-type:50001 > sound:boom
cigi-entity-type:50002 > sound:shot
cigi-entity-type:50003 > sound:engine
cigi-entity-type:50004 > sound:mouseClick
```


4. Send an **Entity Control** packet with the mapping identifier as the Entity Type (for example, 50000), and the **Animation State** parameter = 2 (Play).

   The sound should play with the following conditions:

   - 3D sound effect - Plays in the sound entity's location.
   - Non-positional sound effect - Plays in all locations.

   When the sound stops, or a non-looping sound finishes playing, an **Animation Stop Notification Packet** is sent to the Host. The Host should then destroy the entity. This is a similar process to how particle effects function.

5.2 Automatic Animations

In order to increase the visual fidelity, VBS Blue IG automatically animates certain features on entities:

- Automatic Lifeform Animations (below)
- Automatic Object and Vehicle Animations (below)

5.2.1 Automatic Lifeform Animations

Lifeforms automatically play movement animations based on their forward velocity, stance and equipped weapon; dynamically switching between standing, walking and sprinting.

For controlling behavior using CIGI as a host, see the following packets in the online ICD Reference (https://manuals.bisimulations.com/cigiicd/icd_23_2/index.html):

- **Rate Control**
- **Velocity Control**
- **Entity Lifeform State Component Control**

5.2.2 Automatic Object and Vehicle Animations

Non-lifeform entities such as platforms and static objects (for example, buildings) may simulate the following automatic animations.

> **NOTE**
> Only dynamically created objects animate. Objects that are a part of the terrain do not animate.

Animation	Details
Wheel and Track Rotation	Animates the wheels and tracks based on the vehicle’s forward velocity. For controlling behavior using CIGI as a host, see the following packets in the online ICD Reference (https://manuals.bisimulations.com/cigiicd/icd_23_2/index.html): • **Rate Control** • **Velocity Control**

Animation	Details
Steering	Animates the steering axle of vehicles based on the vehicle's yaw angular velocity. For controlling behavior using CIGI as a host, see the following packets in the online ICD Reference (https://manuals.bisimulations.com/cigiicd/icd_23_2/index.html): • **Rate Control** • **Velocity Control** When using VBS4 as a host, angular velocity is automatically applied.
Suspension	Suspension travel for wheels is simulated based on their distance to the ground.
Time	Provides **simulation** and **clock time** for animations, such as spinning radar dishes or clocks. The **simulation time** is determined based on the *Environment Simulation Time*. It controls the `time` animation source. The **clock time** is determined using the current *Time of Day*. It controls the `clockHour`, `clockHour24`, `clockMinute`, and `clockSecond` animation sources. A very rudimentary time zone conversion based on the entity's current longitude is used to populate the `clockHour` and `clockHour24` animation sources with the local time. For more information on how a simulation host can set these values, see Time and Time Progression (on page 169).
Wind Global	Updates wind-based animations (`windDirection` and `windSpeed`) on objects such as wind socks based on the current global wind velocity value.

By default the automatic animations are enabled. It may be necessary to disable the animations if you wish to fully control an entity's animations with your host. Disabling the animations may also slightly improve performance.

There are several ways to either enable or disable automatic animations. The system is hierarchical, with the lower levels being able to override what is set by the upper levels:

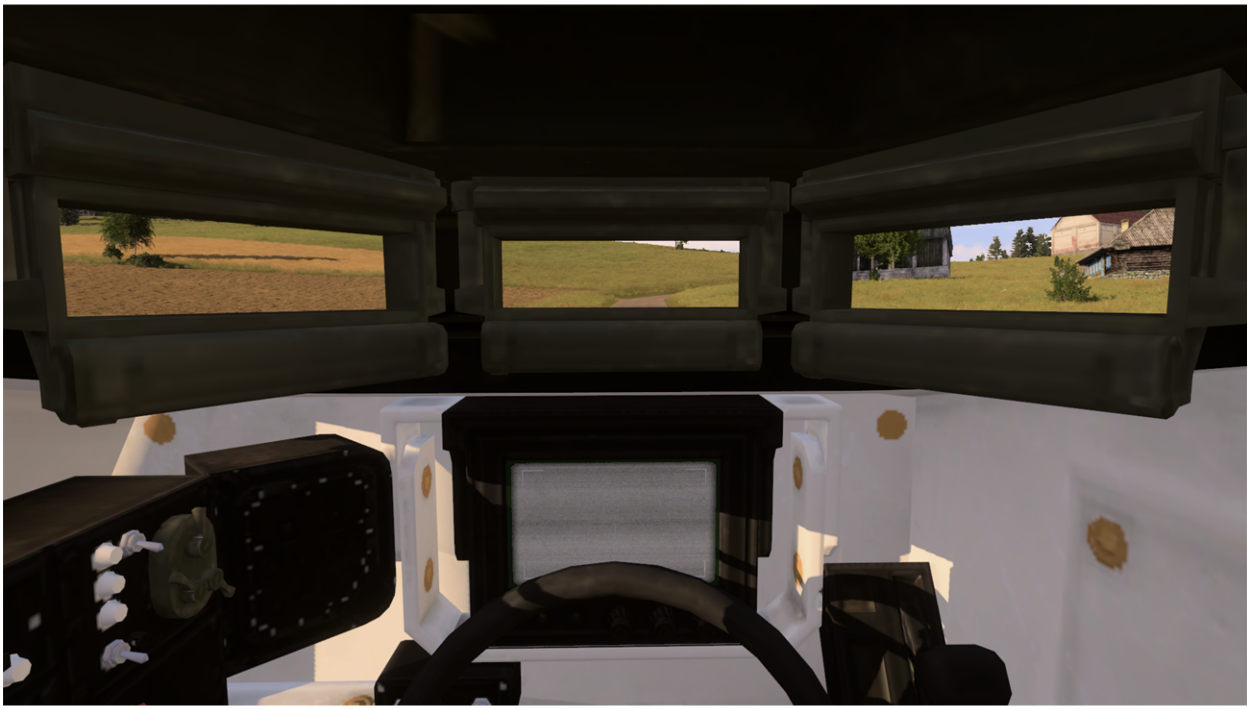
1. **Global setting** - See Entities Settings (on page 238).
2. **CIGI per-session override** - See the **System Global Simulated Animations Component Control** packet in the online ICD Reference (https://manuals.bisimulations.com/cigiicd/icd_23_2/index.html).
3. **CIGI per-entity override** - See the **Entity Simulated Animations Component Control** packet in the online ICD Reference (https://manuals.bisimulations.com/cigiicd/icd_23_2/index.html).

5.3 Mirrors and Periscopes

Set up mirrors and periscopes for relevant vehicle models.

Mirrors provide additional views which appropriately mirror the scene. This re-renders the scene in additional drawing passes.

Periscopes allow the crew to look outside from within the vehicle, efficiently rendering the interior and exterior scene in a single drawing pass.



Mirror and periscope visualization is automatically enabled when certain conditions are met within a specified IG view, specifically when the view is attached to a crew location in a vehicle.

Set up mirror and periscope viewing following the procedure in either of the following options, as required:

- Set Up Mirrors and Periscopes Using VBS as a Host (below)
- Set Up Mirrors and Periscopes Using CIGI as a Host (on page 154)

If using mirrors, additional configuration may be necessary given their impact on performance. For more information, see Mirrors Configuration (on page 156).

5.3.1 Set Up Mirrors and Periscopes Using VBS as a Host

> **NOTE**
> Mirrors and periscopes follow the same set up procedure.

To set up mirrors and periscopes with VBS as a host, an **IG View Object** must be configured that will attach the view to a seat within a vehicle.

Attaching a view to a seat requires the `<CrewPosition>` and `<ParentDrawMode>` tags within the **IG View Object** XML file:

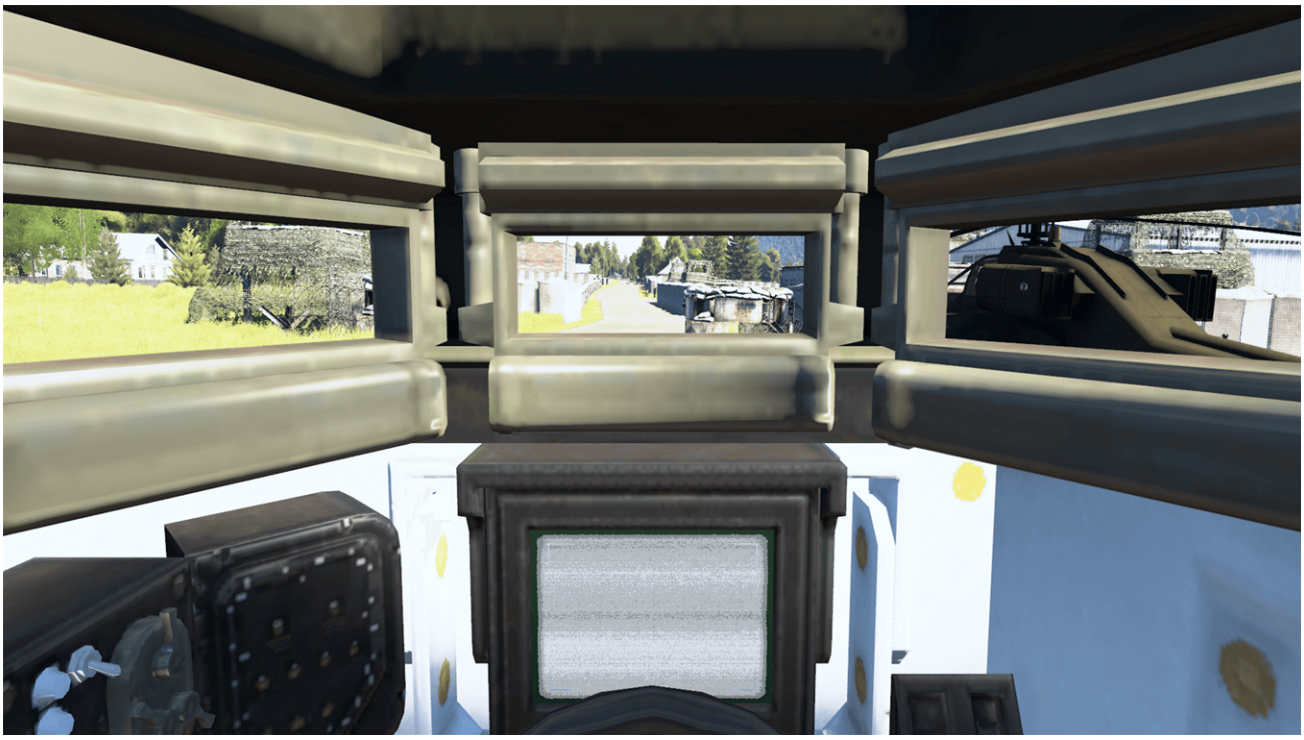
- `CrewPosition` - Numeric value that determines to which seat the view attaches, for example, a Driver or Gunner.
- `ParentDrawMode` - Numeric value that determines the Draw Mode that the parent entity uses when the **IG View Object** is attached to it.

The example code below illustrates an example **IG View Object** attached to the Driver crew position of the vehicle entity **vbs_us_army_stryker_m1126_icv_m2_wdl_x**.

```
<?xml version="1.0"?>
<View_Config>
  <ViewGroup>
    <ID>1</ID>
    <Yaw_Offset>0</Yaw_Offset>
    <Pitch_Offset>0</Pitch_Offset>
    <Roll_Offset>0</Roll_Offset>
    <X_Offset>-0.1</X_Offset>
    <Y_Offset>-0.7</Y_Offset>
    <Z_Offset>0.7</Z_Offset>
    <CrewPosition>0</CrewPosition><!-- 0 is the Driver -->
    <ParentDrawMode>3</ParentDrawMode><!-- 3 enables Cargo Draw Mode -->
  </ViewGroup>
  <View>
    <ID>1</ID>
```

```
    <GroupID>1</GroupID>
    <FOV_Left>-45</FOV_Left>
    <FOV_Right>45</FOV_Right>
    <FOV_Bottom>-25</FOV_Bottom>
    <FOV_Top>25</FOV_Top>
    <FOV_Near>0</FOV_Near>
    <FOV_Far>0</FOV_Far>
    <Yaw_Offset>0</Yaw_Offset>
    <Pitch_Offset>0</Pitch_Offset>
    <Roll_Offset>0</Roll_Offset>
    <X_Offset>0</X_Offset>
    <Y_Offset>0</Y_Offset>
    <Z_Offset>0</Z_Offset>
  </View>
</View_Config>
```

The image below shows the view through the vehicle after implementing the code sample.



NOTE

If multiple viewports are used, mirrors render based on the *top-most* source view by default. This can be configured with the **Mirror Source View Selection Method** property in `DefaultViewConfig.xml`. For more information, see Views and Render Targets (on page 183).

5.3.2 Set Up Mirrors and Periscopes Using CIGI as a Host

NOTE

Mirrors and periscopes follow the same set up procedure.

Setting up mirrors and periscopes with CIGI as a host requires attaching a view to a crew position of a vehicle. The example below illustrates how to do this using **CIGI Test Host**.

Follow these steps:

1. In the `\tools\CigiTools\` directory, run CigiTestHost.exe.

   The CIGI Test Host home page opens. For more general information on using CIGI Test Host, see CIGI Test Host - Overview (on page 21).

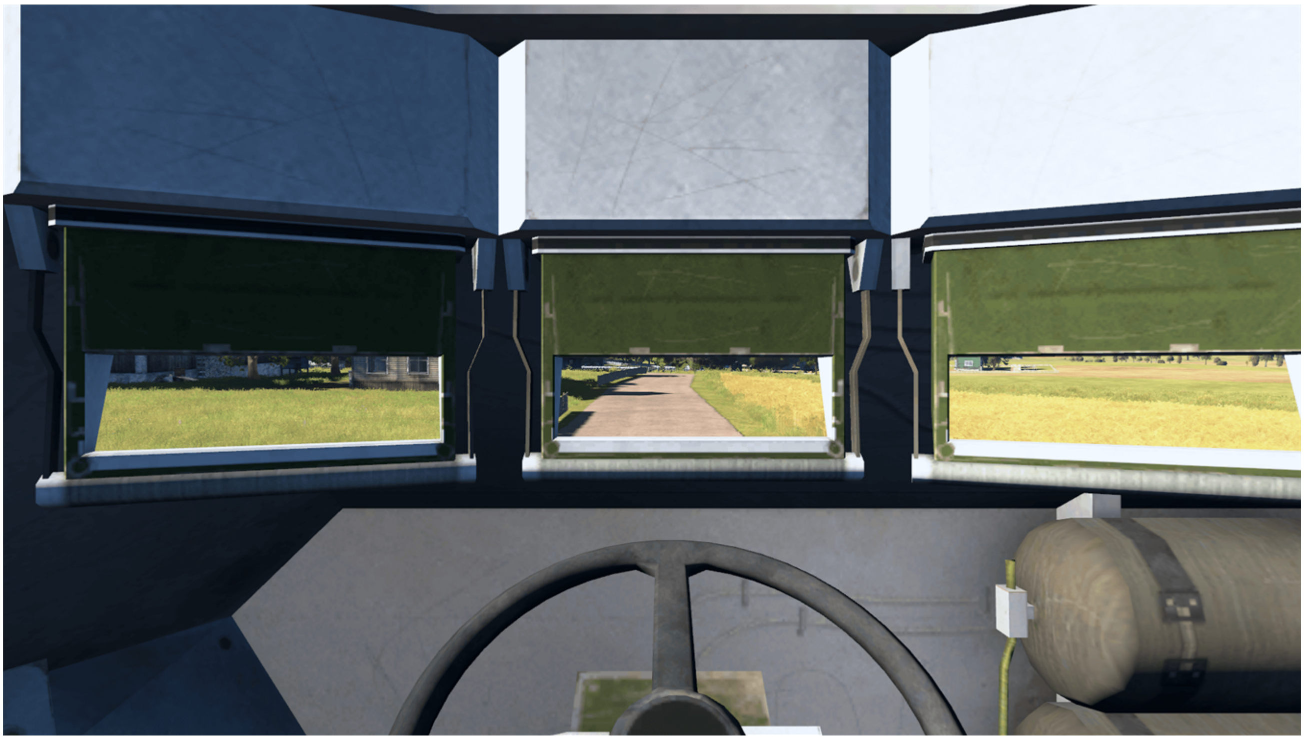
2. On the home page of CIGI Test Host, click **Send Packets**.
3. Click **Load Queue**, and browse to:

   `\tools\CigiTools\resources\SamplePacketQueues\Periscopes.pq`

   To understand the process required, take note of the CIGI packets in the queue.

4. Click **Send Packet Queue**.

   Once the steps above are completed, the View should look like the following:



NOTE

If multiple viewports are used, mirrors render based on the *top-most* source view by default. This can be configured with the **Mirror Source View Selection Method** property in `DefaultViewConfig.xml`. For more information, see Views and Render Targets (on page 183). Alternatively, this can be dynamically changed by the CIGI host with the View Mirror Source View Selection Method Component Control packet.

5.3.3 Mirrors Configuration

Mirrors can heavily impact performance, especially in vehicles with multiple mirrors. The impact on performance can be configured in **Video Settings** settings, using two options in particular:

- **IG / Mirror Fidelity**
  - **Low** - The mirrors render with the following overrides to the global video settings:
    - DLSS = Ultra-Performance or Render Resolution = 50%
    - MSAA = 1x
    - Shadows = Off
    - 25% scene draw distance
    - 50% terrain detail
    - Renders of Sky, Sun, Moon, Stars, Ground, Water, Biome Trees, Biome Bushes, Geometry, Point Clouds, and Objects.
  - **Medium** - The mirrors render with the following overrides to the global video settings:
    - DLSS = Balanced or Render Resolution = 75%
    - MSAA = 1x
    - Shadows = Off
    - 50% scene draw distance
    - 75% terrain detail
    - Renders of Sky, Sun, Moon, Stars, Ground, Water, Biome Trees, Biome Bushes, Geometry, Point Clouds, and Objects.
  - **High** - The mirrors render with the same settings as current global video settings, and with nothing overridden.

> **NOTE**
> If necessary, the fidelity values can be modified in the following file:
> `\IG_Installation\data\BlueProduct\res\VideoSettings\MirrorFidelity.xml`

- **Renderer / RTT Per Frame** - Controls how many render-to-texture targets are drawn each frame. If a vehicle contains multiple mirrors, a value of 1 renders each mirror on separate frames. This results in the best overall performance; however, the mirrors do not render smoothly and "stutter". A value of 8 renders up to 8 render-to-textures per frame, so if a vehicle has 3 mirrors, this renders all mirrors every frame, resulting in worse performance but smoothly updating mirrors.

5.4 Ropes

VBS Blue IG supports simulation and rendering of a new type of object: ropes. A rope can be represented in either of the following ways:

- A catenary between two control entities with a length parameter.
- A cubic Bézier curve between two control entities with an additional two control entities providing the control points.

The entities used for controlling the shape of ropes are typically invisible entities, but they can be any entities positioned in the scenario. As the control entities are moved in the scenario, the rope shape controlled by the entities is updated, as well.

Additionally, ropes have properties that can be customized and include the following:

- Color
- Thickness
- Count of segments

Ropes can be created and controlled via CIGI packets. For more information, see **Entity Rope Control Component Control** in the online ICD Reference (https://manuals.bisimulations.com/cigiicd/icd_23_2/index.html).

Adjust **Ropes** settings for a simulation host using either of the following methods:

- **Settings UI** - Press **Tab** to open the VBS Blue IG Settings (on page 201) and select the **Simulation** option. Click **Ropes** and make adjustments, as required.
- **Edit XML file** - Open the `Simulation.xml` file located in the following directory and edit the Ropes section, as required:

  ```
  %LOCALAPPDATA%\Bohemia Interactive Simulations\
  VBS Blue IG\<version>\Settings\
  ```

  For more information, see Simulation Settings (on page 285).

5.5 Symbology

This section provides details about the Symbology implementation in VBS Blue IG.

- CIGI Symbology (on the next page) - Control Symbology using standard and custom CIGI packets.
- Symbology Lua Scripts (on page 165) - Control Symbology using Lua scripting.

5.5.1 CIGI Symbology

Symbology in VBS Blue IG is fully implemented according to the CIGI 3.3 and 4.0 Specifications. Furthermore, additional drawing primitives are supported to provide Host creators a greater level of control to represent advanced symbology interfaces. For non-CIGI standard types, the host must clone these types from a symbol template defined in the active Symbology configuration using a Symbol Clone packet.

Multiple symbology configurations can be defined in a VBS Blue IG runtime.

The VBS Blue IG installer creates a default `Config.xml` file in a **Default** directory, and uses **Config ID = 0**).

Additionally, the installer creates a sample `Config.xml` file in a **Sample** directory, and uses **Config ID = 1**)

The **Config/@ID** attribute defines the ID used for the Symbology configuration. This ID should be unique across all configurations.

> **NOTE**
> If multiple configurations share the same ID, the behavior is undefined.

Create a symbology configuration with the following procedure.

5.5.1.1 Symbology Configuration

Follow these steps:

1. Create a subdirectory for each symbology configuration in the following directory:

   `\IG_Installation\Components\CigiProtocol\Configs\`

   > **NOTE**
   > For organizational purposes, you may name the sub-directory according to your preference.

2. In the sub-directory, create a `Config.xml` file.

   Example:

   ```
   <Config ID="1"> <!-- Sample CIGI Config -->
       <Atlas Path="Configs/Sample/Atlas/TextureAtlas.xml" />
       <Fonts Path="Configs/Sample/Fonts.def" />
       <Surfaces Path="Configs/Sample/Surfaces.xml" />
       <SymbolTemplates Path="Configs/Sample/SymbolTemplates.xml" />
   </Config>
   ```


Users are expected to make a new configuration folder, for example, **MyConfig**. Other real-world examples could include FLEX-air, FLEX-F18Pilot, FLEX-F18WSO, FLEX-AH64D-Gunner. Each configuration folder must contain a `Config.xml` with a unique ID, such as 10, 11, 12.

The Default configuration is loaded automatically at the beginning of the application startup, and is set up to match the CIGI 3.3 and 4.0 Specification level of support. The Sample configuration provides an example of how to configure each non-standard surface and symbol.

Use the **Symbology Config** packet (see the online ICD Reference (https://manuals.bisimulations.com/cigiicd/icd_23_2/index.html)) to switch the IG to use a specific Symbology configuration.

There are 4 parts to a Symbology configuration:

- Atlas (below)
- Fonts (on the next page)
- Surfaces (on the next page)
- SymbolTemplates (on the next page).

See the `\Sample\Config.xml` file for an example of how each part is defined for each configuration ID. Paths in this file are relative to the base directory:

`\IG_Installation\components\CigiProtocol\`

Explore the following configuration sections.

5.5.1.2 Atlas

A Symbology atlas file is a collection of images, animated images, image masks, and baked font textures, which are combined into large textures for optimized rendering performance. An atlas consists of a single XML file, and one or more texture files.

It is important to note that images, animated images, image masks, and fonts are not read dynamically from their individual files during runtime. They are only read from the texture atlas defined in the active symbology configuration. If such a symbol is not found in the texture atlas, it is not visible, even if the source file exists in the specified location.

To create a texture atlas from source images, the BIAtlasPackerGUI tool is used:

`\IG_Installation\components\SymbologyComponent\Tools\BIAtlasPackerGUI.exe`

In order to repack the Sample texture atlas, use the following values in the tool:

- Base Path:

  `\IG_Installation\components\CigiProtocol\`

- Symbol Templates:

  `\IG_Installation\components\CigiProtocol\Configs\Sample\SymbolTemplates.xml`

- Fonts Definitions:

  `\IG_Installation\components\CigiProtocol\Configs\Sample\Fonts.def`

- Output Directory:

  `\IG_Installation\components\CigiProtocol\Configs\Sample\Atlas\`

This overwrites the existing `TextureAtlas.xml` and `TextureAtlas0.dds` files, with the source images, animated images, image masks, and fonts defined in the `SymbolTemplates.xml` and `Fonts.def` file.

5.5.1.3 Fonts

Fonts are True Type Fonts (`.ttf`) that are embedded in a texture atlas, and mapped to a CIGI Font ID in the `Fonts.def` file.

The format of the `Fonts.def` file is documented at the top of the file:

`\IG_Installation\components\CigiProtocol\Configs\Default\Fonts.def`

> **NOTE**
> The relative font path must be accurately defined relative to the base path used, when generating the texture atlas.

5.5.1.4 Surfaces

Surface configuration allows for definition of templates, which can be applied by CIGI Component Control packets.

See the `\Sample\Surfaces.xml` file for an example on definitions.

5.5.1.5 SymbolTemplates

Symbol Templates are used when using the **Symbol Clone** packet (see the online ICD Reference (https://manuals.bisimulations.com/cigiicd/icd_23_2/index.html)) when cloning from a Symbol Template.

See the `\Sample\SymbolTemplates.xml` file for an example on definitions. The ID attribute on each node represents the Source ID parameter in the Symbol Clone packet.

Symbols

Type	CIGI Standard	Description
Text	Yes	Identical to Symbol Text from CIGI.
Point	Yes	Identical to Symbol Line, Primitive Type = Point from CIGI.

Type	CIGI Standard	Description
Line	Yes	Identical to Symbol Line, Primitive Type = Line, Line Strip, Line Loop from CIGI.
Triangle	Yes	Identical to Symbol Line, Primitive Type = Triangle, Triangle Strip, Triangle Fan from CIGI.
Circle	Yes	Identical to Symbol Circle from CIGI.
Image	No	The symbol is an image (with alpha transparency supported). The image must be baked into a texture atlas to be rendered.
AnimatedImage	No	The symbol is an image (with alpha transparency supported), with sprite atlas based frames that can be switched between. The image must be baked into a texture atlas to be rendered.
Scripted	No	The symbol is a collection of other symbols (such as text, lines, images), that is combined into a single symbol for reduced overhead of manipulating large, static symbol collections. A Lua script is used to configure the collection. We recommend enabling Symbology logging when creating Scripted symbols in order to catch errors and print statements. `\IG_Installation\components\SymbologyComponent\Settings\Symbology.xml` Log/Type = 10 and Log/Level = 3
SVG	No	The symbol is drawn using an SVG file. Note that SVG support is limited in functionality (specifically, only paths are supported).
Mask	No	The symbol is an image (with alpha transparency supported), of which the alpha value is used to mask all other symbols on the same surface layer. The image must be baked into a texture atlas to be rendered. **NOTE** This type is deprecated. It is recommended to instead use the **Symbol Mask Component Control** packet (see the online ICD Reference (https://manuals.bisimulations.com/cigiicd/icd_23_2/index.html)) to enable any symbol to act as a mask.
Browser	No	The symbol is a Chromium based web browser that is directed to load a web page. Web pages can be external (for example, `https://www.google.com`) and require an internet connection, or locally defined using the `client://` scheme, which are located in the `\Domains\` directory (a domain is either a zip file - *`domain.zip`*, or a folder - `\domain\`): `\IG_Installation\components\SymbologyComponent\Domains\`

Type	CIGI Standard	Description
View	No	The symbol is a view that is rendered to the symbol. This allows for sensor / gun camera style views in a vehicle, or picture in picture views. A view is a standard CIGI view that is defined and positioned by the View Definition and View Control packet.

5.5.2 Symbology Lua Scripts

Symbology in VBS Blue IG supports Lua scripting to define and control Symbology. The scripts run using LuaJIT, and the full functionality of Symbology is exposed to Lua scripts, as well as additional functions to extract information from VBS Blue IG to display using Symbology.

Sample Lua scripts are provided in the following directory:

`\IG_Installation\Components\SymbologyComponent\Scripts\Sample\`

- Initializing a script (below)
- Unloading or Reloading a script (on the next page)
- Controlling a script (on the next page)
- Script API (on page 167)

5.5.2.1 Initializing a script

There are multiple ways to initialize a Symbology Lua Script, depending on the type of Host used:

- Configure Settings / Symbology / Scripting / Symbology Lua Scripts to contain the path to a valid Symbology Lua Script. It will then be initialized when VBS Blue IG starts.
  - This allows for a script to be initialized without the need for a CIGI or VBS4 Host. The script can still be interacted with by the appropriate CIGI or VBS4 SQF functions, however, to allow for dynamic Host-defined behavior.
- With a CIGI Host, send a **BISIM Script Message** packet (see the online ICD Reference (https://manuals.bisimulations.com/cigiicd/icd_23_2/index.html)) with Script Type = 0 (Symbology Lua Script Load).
  - Although Symbology can already be controlled through CIGI, it may be simpler and improve performance to use Symbology Lua Scripts to avoid sending hundreds or thousands of CIGI packets each frame for advanced Symbology.
- With a VBS4 Host, execute the `IG_SymbologyLuaScriptLoad` SQF function.
- With **IG SDK** , using the `SymbologyAPI::ScriptLoad` function.

> **NOTE**
>
> The path provided to initialize a script can be an absolute path, or a relative path to the following directory:
>
> `IG_Installation\Components\SymbologyComponent\Scripts\`
>
> For example: `\Scripts\Sample\BasicHud.lua`

When a script is initialized, the following Lua functions in the script will be called automatically, if they are defined in the global context:

- `Init()` when the script is initialized. This can be used to initialize the Symbology context and other Symbology.
- `Shutdown()` when the script is shutting down. This can be used to remove any created Symbology contexts (however this will be done automatically if not explicitly called).
- `Update(dt)` is called each frame, with the `dt` parameter representing the frame delta time in seconds. This can be used to apply changes to the created Symbology, to allow for fully dynamic and reactive displays.

5.5.2.2 Unloading or Reloading a script

A script can be unloaded in multiple ways. This can be useful to dynamically enable or disable a script, as well as reload a script when applying changes:

- Using Symbology (on page 409) Scripts, which contains buttons to unload and reload a script.
- With a CIGI Host, send a **BISIM Script Message** packet (see the online ICD Reference (https://manuals.bisimulations.com/cigiicd/icd_23_2/index.html)) with Script Type = 1 (Symbology Lua Script Unload).
- With a VBS4 Host, execute the `IG_SymbologyLuaScripUnLoad` the SQF function.
- With **IG SDK** , using the `SymbologyAPI::ScriptLoad` function.

5.5.2.3 Controlling a script

A script can be interacted with from an external Host, allowing for additional information to be injected into the script:

- With a CIGI Host, send a **BISIM Script Message** packet (see the online ICD Reference (https://manuals.bisimulations.com/cigiicd/icd_23_2/index.html)) with Script Type = 2 (Symbology Lua Script Execute).
- With a VBS4 Host, execute the `IG_SymbologyLuaScriptExecute` SQF function.
- With **IG SDK** , using the `SymbologyAPI::ScriptExecute` function.

The Lua code specified will be executed in the global Lua context. Consider the following Lua script snippet:

```
function MyGlobalFunction(text)
  print(text)
end

local function MyLocalFunction(text)
  print(text)
end
```


In this example, the `MyGlobalFunction` can be called by executing `MyGlobalFunction("Hello World")`, and "Hello World" would be printed to the Symbology Info Log. However, trying to do the same with `MyLocalFunction` would throw an error, as it is a local function and does not exist in the global context.

This allows the Host to dynamically interact with the script, such as changing the text displayed in a text symbol, or changing the visibility of a collection of symbols. While the method to do this is up to the designer of the script, since CIGI and VBS4 Host should ideally not be sending large amounts of Lua code, a suggested way to have dynamically controlled Symbology is the following:

1. Define a global `SetVars(table)` function.

   This function will set local variables to the values specified by the parameters.

2. In the `Update` function, depending on the state of the local variables, update the Symbology as appropriate.

The `BasicHud.lua` sample script provides an example of this method, where a Host can call `SetVars({compassVisible=false})` to hide the compass, or `SetVars({crosshairVisible=false,compassVisible=false})` to hide both the crosshair and the compass. Depending on the amount of variables that need to be controlled, and to save on bytes sent over the network, multiple domain specific `SetVars` functions and shorter variable names could be used as necessary.

5.5.2.4 Script API

The Symbology Lua Script has full access to the available Symbology API, as well as additional functions to query information available in VBS Blue IG.

The available API is fully documented with Lua annotations, and is provided in the following directory:

`\IG_Installation\Components\SymbologyComponent\Scripts\definitions\`

To ease script development, it is recommended to use Visual Studio Code, with the Lua extension provided by sumneko, which can use the provided annotations to provide an improved editor experience.

5.6 Thermal Simulation Control

Thermal values can be controlled either directly through a Host simulation or automatically calculated within VBS Blue IG with no external simulation control required.

5.6.1 Internal VBS Blue IG Simulation

VBS Blue IG has an internal simulation of thermal parameters that can be enabled through settings. Additionally, Simulation properties can be adjusted to configure how thermal properties are calculated.

Follow these steps:

To enable automatic Thermal simulation

1. Open the settings file:

   `%LOCALAPPDATA%\Bohemia Interactive Simulations\VBS Blue IG\<Version>\Settings\Simulations.xml`

2. Set the parameter `<Simulation Enabled="false">` to `<Simulation Enabled="true">`.

   This parameter enables thermal simulation, allowing entities to heat up and cool down dynamically.

A description for each thermal simulation setting can be found within the settings file.

5.6.2 External Control

External control for thermal parameters is exposed through the following CIGI Packets:

- **Entity Thermal Factors Component Control**
- **Entity Vehicle State Component Control**

For more information, see the online ICD Reference (https://manuals.bisimulations.com/cigiicd/icd_23_2/index.html).

5.7 Time and Time Progression

VBS Blue IG provides control over several independent time systems. Each time can either be controlled directly by the simulation host, or automatically advanced by VBS Blue IG at a host-controlled rate. The systems are the following:

- Time of Day (below)
- Simulation Time (below)

For information about how to control the time within these systems, see Time Control (on the next page).

5.7.1 Time of Day

The time of day value controls the UTC time of the universe. This controls the day and night cycle, star field, sun and moon position, and moon phases.

The current time value can be viewed in **Debug UI (on page 376) / Environment / Current Time Of Day**.

5.7.2 Simulation Time

Simulation time is further split up into the following individual time systems:

- IG Simulation Time (below)
- Environment Simulation Time (below)
- Environment Effect Time (on the next page)
- Environment Adaptation Time (on the next page)

5.7.2.1 IG Simulation Time

The simulation time controls IG specific simulation properties, such as extrapolation, animations, particle effect lifetime, automatic thermal simulation, vehicle light indicators, and other properties related to automated systems in the IG.

The current time value can be viewed in Debug UI (on page 376) **/ Task Director / General / Simulation**.

5.7.2.2 Environment Simulation Time

The environment simulation time controls environment specific simulation properties, such as wind effects, water animation, particle effects, and other properties related to the automated systems in the environment rendering engine.

The current time value can be viewed in Debug UI (on page 376) **/ Environment / Environment Simulation Time**.

5.7.2.3 Environment Effect Time

The environment effect time controls environment specific effect properties, such as precipitation, droplets, sun flare animation, sensor noise, and other properties related to the automated effect systems in the environment rendering engine.

The current time value can be viewed in Debug UI (on page 376) **/ Environment / Environment Effect Time**.

5.7.2.4 Environment Adaptation Time

The environment adaptation time controls environment specific adaptation properties, such as eye adaptation, automatic thermal imaging adjustment, and other properties related to the automated adaptation systems in the environment rendering engine.

The current time value can be viewed in Debug UI (on page 376) **/ Environment / Environment Adaptation Time**.

5.7.3 Time Control

For controlling time using CIGI as a host:

- The IG Control packet can be used to set all simulation times at once, that is, Simulation Time, Environment Simulation Time, Environment Effect Time, and Environment Adaptation Time.
    - **IG Control** - *Timing Value*
- The Environment or Celestial Sphere packets can be used to set the Time Of Day.
    - **Environment Control** - *Hour, Minute, Date*
- Each simulation time value can be individually controlled, as well as their individual progression rates, using the **System Time Component Control** packet.
- The Time Of Day progression rate can be modified using the **Celestial Sphere Time of Day Component Control** Packet.

For more information, see the applicable Packet topics in the online ICD Reference (https://manuals.bisimulations.com/cigiicd/icd_23_2/index.html).

When using VBS4 as a host, the **Time of Day** and **Environment Simulation Time** values are automatically synchronized with the host.

5.8 Tracks and Trails

Vehicle Tracks, Dust Trails, Snow Trails, and Lifeform Tracks are related features in VBS Blue IG, and can enhance the naturalism of the scenario. The following topics provide details about the tracks and trails implementation:

- Vehicle Tracks and Effects (on page 178) - Vehicle tracks settings, which include dust trail effects behind moving vehicles and rotor wash effects, can be controlled globally and individually.
- Lifeform Tracks (on the next page) - The lifeform tracks system creates footstep effects behind moving units.
- Snow Trails (on page 175) - VBS Blue IG generates snow trails (volumetric snow deformations) for moving lifeforms and vehicles by propagating them from a simulation host or by calculating them independently for CIGI entities.

5.8.1 Lifeform Tracks

The lifeform tracks system creates footstep effects behind moving lifeform entities (human and animals). Tracks are drawn when walking and running while standing and crouching, but not while prone. Walking forwards and backwards, and rotation of tracks matches that of the lifeform. No tracks are drawn when the unit only rotates in one place. Other attributes of lifeform tracks include the following:

- Tracks are drawn according to set values.
- Segment count and size affect the drawn tracks.
- Tracks are disabled by default.
- Settings can be customized globally or individually.

> **NOTE**
> Enabling lifeform tracks may cause a drop in performance, especially in scenarios with large entity counts. It is not recommended to be enabled in performance limited situations.

Globally enable or disable the lifeform tracks system with the following settings.

Follow these steps:

1. Press **Tab** to open the VBS Blue IG Settings (on page 201).
2. In the left pane of the menu, select **Entities**.
3. Click the **Lifeforms > Tracks** arrows in the right pane.
4. Enable, disable, or customize Lifeform Tracks generation, as required.

   For more information about these additional options, see Entities / Lifeforms / Tracks settings.
5. Click **Save Settings** in the left pane.

If lifeform tracks are enabled, more precise configuration can be done depending on the Host type:

- VBS as Host (below)
- CIGI as Host (below)
- CIGI Component Control (on the next page)

5.8.1.1 VBS as Host

Configure lifeform tracks from a VBS4 host with the following settings.

1. Press **Tab** to open the VBS Blue IG Settings (on page 201).
2. Select **VBS External Networking** in the left pane of the menu.
3. Click the **Interop** arrow.
4. Click the **Lifeform Tracks** arrow.
5. In the dropdown, select from the **Tracks State** options:
    a. Forced Off
    b. Use Global Settings
    c. Auto

    For more information about these additional options, see VBS External Networking / Interop / Lifeform Tracks settings in the VBS Blue IG Manual.
6. Click **Save Settings** in the left pane.

5.8.1.2 CIGI as Host

Configure lifeform tracks creation for CIGI entities.

Follow these steps:

1. Use the Settings UI.
2. Press **Tab** to open the VBS Blue IG Settings (on page 201).
3. Select **CIGI** in the left pane of the menu.
4. Click the **Session #** arrow.
5. Click the **Lifeform Tracks** arrow.

6. In the dropdown, select from the **Tracks State**options:
    a. Forced Off
    b. Use Global Settings
    c. Auto

    For more information about these additional options, see CIGI / Sessions / Session / Lifeform Tracks settings in the VBS Blue IG Manual.
7. Click **Save Settings** in the left pane.

5.8.1.3 CIGI Component Control

Lifeform tracks can be also controlled for individual entities using the following CIGI component controls:

- Entity Lifeform Tracks Component Control
- Session Settings Component Control

5.8.2 Snow Trails

The snow trails system leverages the features of the volumetric snow implementation and creates real-time deformations of the snow covering in response to entities moving across the scene. Snow is deformed around moving entities with the following parameters in mind:

- **Entity size** - Snow deformations visually correlate with the size of the entity generating the snow trails.
- **Entity stance** - With lifeforms, snow trails generated by standing and crouching units visually differ from those left by prone units.
- **Snow covering height** - The higher the snow coverage, the deeper the snow trail.
- **Snow density** - The higher the snow coverage density, the less pronounced the snow trails.

Creation of new snow surface deformations can be enabled or disabled in the following settings: Simulation / Terrain Deformation (on page 311).

Global behavior of the snow trails system is configured in the following setting: Entities / Lifeforms / Snow Trails (on page 243).

If snow trails are enabled globally, more precise configuration can be done according to one of the following Host types:

- VBS as Host (below)
- CIGI as Host (on the next page)
- CIGI Enhancements (on page 177)

5.8.2.1 VBS as Host

Enable or disable snow trails propagation from a VBS4 host with the following settings.

Follow these steps:

1. Press **Tab** to open the VBS Blue IG Settings (on page 201).
2. Select **VBS External Networking** in the left pane of the menu.
3. Click the **Interop** arrow.
4. Click **Enable Soft Surface Deformations**.
5. To further configure snow trails creation for lifeforms, select **Lifeform Tracks**.
6. In the dropdown, select from the **Snow Trails State** options:
    a. Forced Off
    b. Use Global Settings
    c. Auto
7. Click **Save Settings** in the left pane.

If the necessary settings are enabled, snow trails generated by lifeforms and vehicles in a VBS4 battlespace are automatically propagated to the scene in VBS Blue IG.

The level of snow compression for these trails depends on the snow density value of the Host, which is also automatically propagated and overrides the VBS Blue IG default value.

5.8.2.2 CIGI as Host

Configure snow trails creation for CIGI entities using the Settings UI.

> **NOTE**
> In VBS Blue IG 23.2, vehicle snow trails are not supported, and are only supported for lifeforms. Vehicle snow trails will be introduced in a future update.

Follow these steps:

1. Press **Tab** to open the VBS Blue IG Settings (on page 201).
2. Select **CIGI** in the left pane of the menu.
3. Click the **Session #** arrow.
4. Click the **Lifeform Tracks** arrow.
5. In the dropdown, select from the **Snow Trails State** options:
    a. Forced Off
    b. Use Global Settings
    c. Auto
6. Adjust snow density according to the level of snow compression, as needed.
    a. In the left pane, select **Simulation**.
    b. Click the **Environment > Global Precipitation** arrows.
    c. In the **Snow Density** slider menu, adjust the level of snow compression from a lower value density to create fresh fallen snow to a higher value density to create solid ice.

    > **NOTE**
    > Snow density can also be controlled with the Global Terrain Surface Snow Component Control packet.

7. Click **Save Settings** in the left pane.

5.8.2.3 CIGI Enhancements

Snow trails can be also controlled for individual entities using the following CIGI component controls:

- Entity Lifeform Tracks Component Control
- Session Settings Component Control

5.8.3 Vehicle Tracks and Effects

Create vehicle tracks, vehicle dust trails, and helicopter rotor wash effects in a scenario. Vehicles can create tracks and dust trail effects trailing behind a moving vehicle, while helicopter rotors can create an appropriately sized and colored wash effect.

Use the following procedures to modify settings:

- Configure Global Settings (below) - Set up tracks and effects on a global level.
- Configure by Host type and Entity (on the next page) - Set up tracks and effects on a per-entity level.

5.8.3.1 Configure Global Settings

- Vehicle Tracks and Effects (above)
- Dust Trail Effect (on the next page)
- Helicopter Rotor Wash (on the next page)

5.8.3.1.1 Vehicle Tracks System

Globally enable or disable the vehicle tracks system with the following settings.

Follow these steps:

1. Press **Tab** to open the VBS Blue IG Settings (on page 201).
2. In the left pane of the menu, select **Entities**.
3. Click the **Vehicles > Tracks** arrows in the right pane.
4. Enable, disable, or customize Vehicle Tracks, as required.

   For more information about these additional options, see Entities / Vehicles / Tracks settings.
5. Click **Save Settings** in the left pane.

Once enabled, continue setup by host type and entity-level control, as required. See Configure by Host type and Entity (on the next page).

5.8.3.1.2 Dust Trail Effect

Globally enable or disable dust trail effects with the following settings.

> **NOTE**
> The effect is sized and colored appropriately based on the velocity of the vehicle, the underlying surface material, and whether the vehicle is contacting the surface or not.

Follow these steps:

1. Press **Tab** to open the VBS Blue IG Settings (on page 201).
2. In the left pane of the menu, select **Entities**.
3. Click the **Vehicles > Dust Trails** arrows in the right pane.
4. Enable, disable, or customize Dust Trails, as required.

   For more information about these additional options, see Entities / Vehicles / Dust Trails settings.
5. Click **Save Settings** in the left pane.

Once enabled, continue setup by host type and entity-level control, as required. See Configure by Host type and Entity (below).

5.8.3.1.3 Helicopter Rotor Wash

Globally enable or rotor wash effects with the following settings.

Follow these steps:

1. Press **Tab** to open the VBS Blue IG Settings (on page 201).
2. Select **CIGI** in the left pane of the menu.
3. Click the **Session #** arrow.
4. Select the **Rotor Wash** arrow and click the box to enable rotor wash by default.

   For more information about this setting, see CIGI / Sessions / Session / Rotor Wash settings in the VBS Blue IG Manual.
5. Click **Save Settings** in the left pane.

Once enabled, continue setup by host type and entity-level control, as required. See Configure by Host type and Entity (below).

5.8.3.2 Configure by Host type and Entity

- VBS as Host (on the next page)
- CIGI as Host (on the next page)

- CIGI Component Controls (on page 182)

5.8.3.2.1 VBS as Host

Configure vehicle tracks, dust trails, or helicopter wash from a VBS4 host with the following settings.

1. Press **Tab** to open the VBS Blue IG Settings (on page 201).
2. Select **VBS External Networking** in the left pane of the menu.
3. In the right pane, choose any of the preferred options:
   - For **vehicle tracks** or **dust trails**.
     a. Click the arrow **Interop > Vehicle Tracks**.
     b. Select from the **Tracks State** or **Dust Trails State** dropdown options.

     For more information about these additional options, see VBS External Networking / Interop / Vehicle Tracks settings in the VBS Blue IG Manual
   - For **helicopter rotor wash**
     a. Click the arrow **General**.
     b. To enable, click **Enable Extra Rotor Effects**.

     For more information about this setting, see VBS External Networking / General / Enable Extra Rotor Effects settings in the VBS Blue IG Manual.
4. Click **Save Settings** in the left pane.

5.8.3.2.2 CIGI as Host

Configure vehicle tracks, dust trails, or helicopter wash with a CIGI host.

Follow these steps:

1. Use the Settings UI.
2. Press **Tab** to open the VBS Blue IG Settings (on page 201).

3. Choose the preferred options:
   - For **vehicle tracks** and **dust trails**:
     a. Select **CIGI** in the left pane of the menu.
     b. Click the **Session # > CIGI** arrows.
     c. Click the **Vehicle Tracks** arrow.
     d. In the dropdown, select from the **Tracks State** or **Dust Trails State** options.

     For more information about these additional options, see CIGI / Sessions / Session / Vehicle Tracks settings in the VBS Blue IG Manual.
   - For **helicopter rotor wash**:
     a. Select **CIGI** in the left pane of the menu.
     b. Click the **Session # > Rotor Wash** arrows.
     c. To enable, click **Enable Rotor Wash Effects By Default**.

     For more information about these additional options, see CIGI / Sessions / Session / Rotor Wash settings in the VBS Blue IG Manual.

     > **NOTE**
     > This procedure also serves as a global configuration for setting helicopter rotor wash.
4. Click **Save Settings** in the left pane.

5.8.3.2.3 CIGI Component Controls

Vehicle tracks, dust trails, and helicopter wash can be controlled for individual entities using the following CIGI component controls:

- **Vehicle tracks and dust trails**
  - Entity Vehicle Tracks State Component Control
  - Session Settings Component Control
- **Rotor wash**
  - Rotor wash can be enabled or disabled per-entity in CIGI, using the Entity Vehicle Tracks State Component Control packet.

> **NOTE**
> The default settings for the rotor wash effect are designed for scenarios using only a few helicopters. If using more helicopters (5+), and to avoid performance impact of the effect, it is recommended to adjust the rotor wash settings, such as **Intensity**, **Loiter Power**, and other settings. Make these adjustments using the Settings / Entities / Rotorwash settings in the VBS Blue IG Manual.

5.9 Views and Render Targets

VBS Blue IG can support multiple views defined in a single instance. In order for the view to be visible and rendered in a location, the view must be configured to define a render method. The view can be configured further to encompass different targets.

The views configuration is shared between all host types, CIGI, and VBS4.

View parameters are set using the `DefaultViewConfig.xml` file, available in:

`\IG_Installation\components\BlueIGViewSystems\Config\`

For more information, see Startup Parameters (on page 454).

If required, you can redirect the path to the xml file using the command line parameter `-defaultViewConfig=<path>`.

Parameters for setting views are divided into the following:

- Views (below) - See this section for guidance on configuring required and optional views parameters.
- Render Targets (on page 194) - See this section for guidance on rendering to different targets within the views parameters.

5.9.1 Views

Parameters for setting views are divided into the following:

- Required Fields (below)
- Optional Fields (on the next page)

It is recommended to configure the view configuration while running VBS Blue IG, using **Debug UI / View Manager / <view>**. Once the parameters have been configured, select **Copy View XML To Clipboard** to copy the appropriate XML structure for the changed parameters. This can then be pasted into the `DefaultViewConfig.xml` file.

5.9.1.1 Required Fields

Views has one required field:

- View ID (below)

View ID

Views require an ID in order to be identified. This ID can be used by external hosts using CIGI through the **View Control** and **View Definition** packets (see the online ICD Reference (https://manuals.bisimulations.com/cigiicd/icd_23_2/index.html)) or in VBS4 by using the **IG View Object**.

Example:


```
<View ID="1">
```

> **NOTE**
> By default, views with `ID="1"` are configured to render to the main target window.

5.9.1.2 Optional Fields

Views have nine optional fields:

- RenderMethod (below)
- Frustum (on page 187)
- SensorType (on page 188)
- SensorParameters<SensorType> (on page 188)
- Viewport (on page 188)
- MirrorMode (on page 188)
- PixelReplicationMode (on page 189)
- AudioListener (on page 189)
- MirrorSourceView (on page 190)
- RenderTarget (on page 190)
- VideoSettings (on page 190)
- ColorRenderFeatures (on page 191)
- DepthRenderFeatures (on page 191)
- GenericWarp (on page 191)
- ScalableWarp (on page 192)
- DomeProjectionWarp (on page 192)
- Final Combined View Example (on page 193)

RenderMethod

This field specifies how a view is rendered or what its role is in an advanced view hierarchy. The following options are available:

- **Screen** - Renders the view to a window or to the entire screen if Fullscreen is enabled for rendering.

- **Fullscreen** - This method is identical to the Screen render method with the following exceptions:
  - Overrides and renders on top of all Screen views
  - Ignores the Viewport parameter and renders to the entire render target.

> **NOTE**
> Fullscreen window rendering does not mean the same as this Fullscreen render method.

- **MultiViewParent** - The view acts as a container to child **MultiView** views. The parent view will not render, but it defines the position, sensor type, and video settings used for all its attached child views.The view position offsets and orientations are used to position the view, and the view viewport and frustum properties are ignored.
- **MultiView** - The view is "attached" to the parent **MultiViewParent**. This view is rendered in an optimized way if the hardware supports multi-view rendering. The view orientations are relative to the parent view, and position offsets are not supported. The view viewport and frustum are used to display the view on the render target. If the total frustum of all child views in the multi-view group is too large, visual anomalies occur. When specifying this render method, the **Parent** attribute must also be defined, specifying the View ID of the **MultiViewParent** to which it is being attached.

> **NOTE**
> **Multi-View rendering** is a special configuration for views. It allows multiple view projections to be rendered in an optimized way, significantly improving performance. However, it has a number of limitations:
>
> - It is only supported on hardware that supports multi-view rendering (NVIDIA Turing based GPUs).
> - A maximum of four views can be attached in a multi-view group.
> - The views can only be oriented slightly from the parent view; offset positions are not supported.
> - Only 1 MultiViewParent can be defined.
> - If the views are in different locations, regular viewports are required. When running multi-view rendering on non-supported hardware, there is still a slight performance improvement due to scene preparation only occurring once for the multi-view group.

- **GroupParent** - Similar to MultiViewParent, this view acts as a container for views that use the **Screen** or **MultiViewParent** or **None** render methods and have their **Parent View ID** attribute pointing to this view. The parent view will not render, but it defines the video settings used for attached child views.

> **NOTE**
>
> **Grouped views rendering** - When 2 or more views share a group parent, they still render individually, but are all prepared together, boosting performance. Grouped views differ from the Multi-View Rendering configuration in the following ways:
>
> - Child views can be in a slightly different location, for example, within a few meters, not completely different.
> - Child views can have different sensor types.
> - A **GroupParent** can have unlimited number of child views, unlike **MultiView** which can only have 4 projections.
> - All views in a group need to be looking within the same ~150-160 degree field of view (accounting for the frustum size as well). Similar to the **MultiView** configuration, if the total view angle of child views in the group is too large, ground or object culling issues may occur.

- **None** - Prevents the view from being rendered anywhere.

Examples:

- **Standard view configuration**

```
<View ID="1">
  <RenderMethod Method="Screen" />
</View>
```

- **Multi-view configuration**

```
<View ID="1">
  <RenderMethod Method="MultiViewParent" />
</View>
<View ID="2">
  <RenderMethod Method="MultiView" Parent="1" />
  <Viewport Left="0.0" Top="0.0" Right="0.5" Bottom="0.5" Layer="0" />
</View>
<View ID="3">
  <RenderMethod Method="MultiView" Parent="1" />
  <Viewport Left="0.5" Top="0.0" Right="1.0" Bottom="0.5" Layer="0" />
</View>
<View ID="4">
  <RenderMethod Method="MultiView" Parent="1" />
  <Viewport Left="0.0" Top="0.5" Right="0.5" Bottom="1.0" Layer="0" />
</View>
<View ID="5">
  <RenderMethod Method="MultiView" Parent="1" />
  <Viewport Left="0.5" Top="0.5" Right="1.0" Bottom="1.0" Layer="0" />
</View>
```

- **Viewport Group configuration**

```
<View ID="1">
  <RenderMethod Method="GroupParent" />
</View>
<View ID="11">
  <RenderMethod Method="Screen" Parent="1" />
  <Viewport Left="0.0" Right="0.5" Top="0.0" Bottom="1.0" Layer="0" />
</View>
<View ID="12">
  <RenderMethod Method="Screen" Parent="1" />
  <Viewport Left="0.5" Right="1.0" Top="0.0" Bottom="1.0" Layer="0" />
</View>
```

Frustum

This field is the frustum used by the view. Near is defined in meters from the camera origin. Left / Right / Top / Bottom are defined in degrees from the center point of the frustum.

Example:

```
<Frustum Near="0.01" Left="-45.0" Right="45.0" Top="30" Bottom="-30"
          Projection="Perspective" />
```

SensorType

This field sets the sensor used by the view. VBS Blue IG options includes `BlankScreen`, `Classification`, `Disable`, `MRTDebug`, `NightVision`, `Normal`, `SAR`, and `ThermalImaging`. These are the string names of the sensor, and any custom sensors can also be used here, as well.

Example:

```
<SensorType Type="Normal" />
```

SensorParameters<SensorType>

This field sets parameters for the specified sensor type. For example, `SensorParametersNormal` for the Normal sensor type. Available sensor parameters can be determined and configured using **Debug UI / View Manager / <view> / Sensor / Parameters**, and differ based on the selected Sensor Type. Once appropriate parameters are configured, use the **Copy View XML To Clipboard** button to get the correct XML structure for the changed parameters.

Example:

```
<SensorParametersNormal>
  <Parameter Name="ColorVision/NoiseDynamic" Value="0.15" />
  <Parameter Name="Tonemap/Exposure" Value="1" />
  <Parameter Name="DepthOfField/Enabled" Value="true" />
</SensorParametersNormal>
```

Viewport

This field defines the area within the Render Target that this view will occupy. Left / Top / Right / Bottom are defined as 0-1 values where 0 is the top left and 1 is the bottom right of the Render Target. The values are a percentage of the Render Target dimensions.

Example:

```
<Viewport Left="0.0" Top="0.0" Right="1.0" Bottom="1.0" Layer="0" />
```

MirrorMode

This field sets mirroring for the view. Available mirror modes include the following:

- None - The view is not mirrored.
- Horizontal - The view is mirrored horizontally.
- Vertical - The view is mirrored vertically.
- HorizontalAndVertical - The view is mirrored horizontally and vertically.

Example:

```
<MirrorMode Mode="None" />
```

PixelReplicationMode

This field sets pixel replication for the view, and can be used to apply a digital zoom type effect for the view. Available pixel replication modes include the following:

- None - The view has no pixel replication.
- 1x2 - The view has every vertical pixel duplicated twice, resulting in a vertically stretched image.
- 2x1 - The view has every horizontal pixel duplicated twice, resulting in a horizontally stretched image.
- 2x2 - The view has every horizontal and vertical pixel duplicated twice, resulting in an image that has a 2x digital zoom.

Example:

```
<PixelReplicationMode Mode="None" />
```

AudioListener

This field defines which view is the audio listener and uses one of two settings:

- Auto - The topmost view automatically determines the audio listener.

  Example:

  ```
  <AudioListener Active="Auto" />
  ```

- Force - Forces a view as the listener.

  AudioListener can have a **Cone node**. By default, the Cone node uses an omnidirectional cone. Otherwise, specific attributes need to be set.

  The attributes correspond to the variables of the X3DAUDIO_CONE struct. For further information, see Microsoft documentation at X3 Audio Cone Structure.

  Unlike the other attributes, if the Cone node is present, all of its attributes are required.

  Example:

  ```
  <AudioListener Active="Force">
    <Cone InnerAngle="1.570796327" OuterAngle="3.141592654" InnerVolume="1.0"
          OuterVolume="0.708" InnerLPF="0.0" OuterLPF="0.25"
          InnerReverb="0.708" OuterReverb="1.0" />
  </AudioListener>
  ```

MirrorSourceView

This field defines which view acts as the source view for mirror rendering. The mirror source view is used when rendering mirrors, and controls the correct view position and orientation of the reflected image. When using multiple views and mirrors, it is important to explicitly control which view is the selected mirror source view. If relying on the Auto selection method, the correct view may not be automatically selected to act as the mirror source view, which results in a wrong view being reflected in the mirror.

The following are the available SelectionMethods:

- **Auto** - The view is configured to use the Auto selection method. If all active views are configured to use the Auto selection method, the mirror source view is selected based on the highest viewport layer, then the highest view ID.
- **Force** - The view is forced as the selected mirror source view.

> **NOTE**
> Only a single view can be forced as the selected mirror source view.

Example:

```
<MirrorSourceView SelectionMethod="Auto" />
```

RenderTarget

This field is the ID of the Render Target for the view. The ID must correspond to a Render Target defined within `<RenderTargets>`.

Example:

```
<RenderTarget ID="0" />
```

VideoSettings

This field sets parameters for a view to override the global video settings on a per-viewport basis. Available per-viewport video settings can be determined and configured using **Debug UI / View Manager / <view> / Video Settings**. Once appropriate parameters are configured, use the **Copy View XML To Clipboard** button to get the correct XML structure for the changed parameters.

Example:

```
<VideoSettings>
  <VideoSetting Name="VideoSettings/Renderer/render_resolution_1" Value="80" />
  <VideoSetting Name="VideoSettings/Renderer/dlss" Value="Disabled" />
  <VideoSetting Name="VideoSettings/Renderer/motion_blur" Value="true" />
</VideoSettings>
```

ColorRenderFeatures

This field configures view specific scene render features enabled for rendering to the color buffer. If not defined (default), the view applies the render features from the global Environment settings. If defined but empty, the view does not render anything to the color buffer.

The available ColorRenderFeature values include the following:

- Sky
- Sun
- Moon
- Stars
- Ground
- Water
- BiomeTrees
- BiomeBushes
- BiomeGrass
- Geometry
- PointClouds
- Objects
- Lights
- Particles
- Clouds
- Precipitation

Example:

```
<ColorRenderFeatures>
  <ColorRenderFeature>Sky</ColorRenderFeature>
  <ColorRenderFeature>Ground</ColorRenderFeature>
  <ColorRenderFeature>Objects</ColorRenderFeature>
</ColorRenderFeatures>
```

DepthRenderFeatures

This field configures view specific scene render features enabled for rendering to the depthbuffer. If not defined (default), the view applies the render features from the global Environment settings. If defined but empty, the view does not render anything to the color buffer.

The available DepthRenderFeature values include the following:

- Ground
- Water
- BiomeTrees
- BiomeBushes
- BiomeGrass
- Geometry
- PointClouds

Example:

```
<DepthRenderFeatures>
  <DepthRenderFeature>Ground</DepthRenderFeature>
</DepthRenderFeatures>
```

GenericWarp

This field sets a view for generic warping. For more information, see Generic Warping Setup (on page 133).

The configuration requires the `-warp` command line parameter to function.

NOTE

Generic Warp does not do any warping or blending itself, and is designed to work in combination with external third-party warp and blend software.

- **Enable** - Enables a view to have its frustum, position offset, and orientation offset overridden by the values specified in the Definition file.
- **Definition** - The path to the definition file. Can either be an absolute path, or a path relative to:

  `\IG_Installation\Components\BlueIGViewSystems\`.

Example:

```
<GenericWarp Enabled="False" Definition="GenericWarp\GenericWarpDefinition.xml" />
```

ScalableWarp

This field sets a view for Scalable warping and blending. For more information, see Scalable Warping Setup (on page 140).

The configuration requires the `-scalable` command line parameter to function.

- **Enable** - Enables a view to be warped and blended using the third-party Scalable Display Manager software.
- **MeshFile** - The path to the `ScalableData.ol` mesh file. Can either be an absolute path, or a path relative to:

  `\IG_Installation\Components\BlueIGViewSystems\`

  Keep blank to automatically find the Scalable mesh file using the search order in **Views/Warping/Scalable** settings.

Example:

```
<ScalableWarp Enabled="False" MeshFile="" />
```

DomeProjectionWarp

This field sets a view for DomeProjection warping and blending. For more information, see DomeProjection Warping Setup (on page 137).

The configuration requires the `-domeprojection` command line parameter to function.

- **Enable** - Enables a view to be warped and blended using the third-party DomeProjection software.
- **ChannelID** - The DomeProjection channel ID to use for this view.

Example:

```
<DomeProjectionWarp Enabled="False" ChannelID="0" />
```

Final Combined View Example

The example code below combines all required and optional fields described.

```
<Config>
  <Views>
    <View ID="[ Required View ID ]">
      <Frustum Near="0.01" Left="-45.0" Right="45.0" Projection="Perspective"
               Top="[ ((#Right-#Left)/CurrentAspectRatio)/2 ]"
               Bottom="[ -((#Right-#Left)/CurrentAspectRatio)/2 ]" />
      <SensorType Type="Normal" />
      <Viewport Left="0.0" Top="0.0" Right="1.0" Bottom="1.0" Layer="0" />
      <MirrorMode Mode="None" />
      <PixelReplicationMode Mode="None" />
      <RenderMethod Method="None" Parent="[Invalid View ID]" />
      <SensorParametersNormal EyeAccommodation="-1"
          AutoEyeAccomMinLuminance="0.226" AutoEyeAccomMaxLuminance="20000"
          AutoEyeAccomTimeRise="0.75" AutoEyeAccomTimeDrop="12"
          AutoEyeAccomBrightness="0.4" AutoEyeAccomFocusCenterX="0.5"
          AutoEyeAccomFocusCenterY="0.5" AutoEyeAccomFocusRadius="0.5224"
          AutoEyeAccomFocusDazzling="5" AutoEyeAccomFocusDazzlingRadius="3"
          Exposure="-0.5" Contrast="-30" Saturation="2" Vibrance="50"
          TonemapperType="5" Cutoff="0.7115052" TemperatureShadow="-1"
          TemperatureHighlight="5" TintShadow="0" TintHighlight="0"
          NoiseDynamic="0" NoiseStatic="0" NoiseSize="1000" NoiseHot="0"
          NoiseHotThreshold="0.99999" Blurring="0" DisplayLuminance="10"
          ScotopicRange="0.1" ScotopicDesaturation="0.5" ScotopicBlueShift="2"
          ScotopicDarkening="3" BloomIntensity="0.05" BloomSpread="0.85"
          ChromaticBloomSpread="1" />
      <SensorParametersNV EyeAccommodation="-1" AutoEyeAccomMinLuminance="0.8"
          AutoEyeAccomMaxLuminance="20000" AutoEyeAccomTimeRise="0.75"
          AutoEyeAccomTimeDrop="12" AutoEyeAccomBrightness="0.4"
          AutoEyeAccomFocusCenterX="0.5" AutoEyeAccomFocusCenterY="0.5"
          AutoEyeAccomFocusRadius="0.5224" AutoEyeAccomFocusDazzling="5"
          AutoEyeAccomFocusDazzlingRadius="3" Gain="3" GainPOW="0.4"
          Blooming="0.078" BloomingPOW="1" BlurringStrength="0"
          BlurringFlareStrength="0" BlurringEdge="0" BlurringCenter="0.5"
          ScopeVignetting="0" ScopeFOV="179.9" ScopeEdgeThickness="0"
          ScopeEdgeHardness="100" Flare1Distortion="0" Flare2Distortion="0"
          Flare3Distortion="0" FlareDistortionCubic="30" FlareIntensity="0"
          CenterIntensity="0" LensDistortionK="0" LensDistortionCubic="0"
          NoiseDynamic="0.005" NoiseStatic="0.005" NoiseSize="1000" NoiseHot="0"
          NoiseHotThreshold="0.99999" SensitivityRed="0.33" SensitivityGreen="0.33"
          SensitivityBlue="0.33" SensitivityNIR="1" FilterIntensityRed="0"
          FilterIntensityGreen="1" FilterIntensityBlue="0" BloomIntensity="0.2"
```



```
            BloomSpread="1" />
        <SensorParametersTI DynamicRange="45" BlackLevel="0" NoiseDynamic="1.195"
            NoiseStatic="2.135" Blur="1.25" BlurEdge="6" LutIndex="0" />
        <AudioListener Active="Auto" />
        <MirrorSourceView SelectionMethod="Auto" />
        <RenderTarget ID="0" />
        <GenericWarp Enabled="False"
            Definition="GenericWarp\GenericWarpDefinition.xml" />
        <ScalableWarp Enabled="False" MeshFile="" />
        <DomeProjectionWarp Enabled="False" ChannelID="0" />
      </View>
    </Views>
</Config>
```

5.9.2 Render Targets

To render to different targets, configure views by using the **RenderTargets** setting. There are currently two types of **RenderTargets**: Window and Offscreen.

- **Window** - Renders the assigned views to a target window.
- **Offscreen** - Renders the assigned views to a texture which can be accessed using the VBS IG SDK rendering interface.

> **NOTE**
> This view is not be visible without using VBS IG SDK.

By default, a single Window render target is created (Render Target ID = 0), and this defines the "Main Window". Additional render targets must have their ID >= 1.

To create other render targets, place the following in the `DefaultViewConfig.xml` file:

```
<Config>
  <RenderTargets>
    <RenderTarget Type="Window" ID="1">
      <Window X="0" Y="0" WindowWidth="1024" WindowHeight="768"
              IsBorderless="False" />
    </RenderTarget>
  </RenderTargets>
</Config>
```

In this example, Render Target ID 1 is a Window with a border, positioned at the top left of the primary monitor, with a size of 1024 x 768 pixels.

To configure View ID 1 to render to the bottom right quadrant of the previously mentioned render target, the following is used:


```
<Config>
  <Views>
    <View ID="1">
      <RenderMethod Method="Screen" />
      <Viewport Left="0.5" Top="0.5" Right="1.0" Bottom="1.0" Layer="0" />
      <RenderTarget ID="1" />
    </View>
  </Views>
</Config>
```

By using a combination of Render Targets and Viewports, multiple views can be configured to be visible for each instance of VBS Blue IG.

In the following example:

- View ID 1 renders to the entirety of the main window render target.
- View IDs 2 through 5 renders to a separate window, positioned at 1920,0, with a window and render size of 1920x1080px. If the primary monitor is at 1920x1080 resolution, and a second monitor is placed to the right of the primary monitor, this positions the window on the secondary monitor (position is relative to the primary monitor top left corner).

  The views are displayed in a 2x2 grid, ordered as:

  - View 2 is placed in the top left quadrant.
  - View 3 is placed in the top right quadrant.
  - View 4 is placed in the bottom left quadrant.
  - View 5 is placed in the bottom right quadrant.
- View ID 6 renders to the entirety an Offscreen Render Target, of size 512 x 512 px for manipulation using VBS IG SDK.

```
<Config>
  <Views>
    <View ID="1">
      <RenderMethod Method="Fullscreen" />
    </View>
    <View ID="2">
      <Viewport Left="0.0" Top="0.0" Right="0.5" Bottom="0.5" Layer="0" />
      <RenderMethod Method="Screen" />
      <RenderTarget ID="1" />
    </View>
    <View ID="3">
      <Viewport Left="0.5" Top="0.0" Right="1.0" Bottom="0.5" Layer="0" />
      <RenderMethod Method="Screen" />
      <RenderTarget ID="1" />
    </View>
```

```
    <View ID="4">
      <Viewport Left="0.0" Top="0.5" Right="0.5" Bottom="1.0" Layer="0" />
      <RenderMethod Method="Screen" />
      <RenderTarget ID="1" />
    </View>
    <View ID="5">
      <Viewport Left="0.5" Top="0.5" Right="1.0" Bottom="1.0" Layer="0" />
      <RenderMethod Method="Screen" />
      <RenderTarget ID="1" />
    </View>
    <View ID="6">
      <RenderMethod Method="Fullscreen" />
      <RenderTarget ID="2" />
    </View>
  </Views>
  <RenderTargets>
    <RenderTarget Type="Window" ID="1">
      <Window X="1920" Y="0" WindowWidth="1920" WindowHeight="1080"
              IsBorderless="False" />
    </RenderTarget>
    <RenderTarget Type="Offscreen" ID="2">
      <Offscreen RenderWidth="512" RenderHeight="512" />
    </RenderTarget>
  </RenderTargets>
</Config>
```

Update Mode

Performance of render targets can be enhanced by using the **UpdateMode** attribute when defining render targets. This option is available for render targets that use either the **Window** or the **Offscreen** type. The **UpdateMode** attribute can have one of two values:

- **Always** - The render target renders every frame.
- **WithBudget** - The render target renders as often as it can, based off the RTT Per Frame video setting. For example, 4 render targets, each with `WithBudget`, with RTT Per Frame = 1, will mean that the render target will render every 4th frame.

> **NOTE**
> The exact update rate also depends on other render targets in the scene, such as mirrors, symbology, etc.

Example of creating two render targets with different update modes:

```
<RenderTargets>
  <RenderTarget Type="Window" ID="1" >
```



```
    <Window X="0" Y="0" WindowWidth="1920" WindowHeight="1080" IsBorderless="False"
UpdateMode="Always" />
  </RenderTarget>
  <RenderTarget Type="Window" ID="2" >
    <Window X="1920" Y="0" WindowWidth="1920" WindowHeight="1080"
IsBorderless="False" UpdateMode="WithBudget" />
  </RenderTarget>
</RenderTargets>
```


6. Runtime Controls

VBS Blue IG provides the following controls:

- UI Controls (below)
- Keyboard and Mouse Controls (below)

6.1 UI Controls

The UI controls are divided into the following:

- VBS Blue IG **Settings** - This menu provides a graphical interface for modifying component settings within VBS Blue IG. For more information, see VBS Blue IG Settings (on page 201).
- **Debug UI** - Use this menu to control VBS Blue IG runtime performance, configuration, and various debug settings. For more information, see Debug UI (on page 376).

6.2 Keyboard and Mouse Controls

The keyboard and mouse controls are divided into the following:

- Fallback Camera Controls (on the next page) - Manually control the camera on VBS Blue IG clients with these commands.
- **Additional Controls** - Additional runtime controls:

Key	Action
Esc	Toggle cursor visibility. **NOTE** Cursor is required to be enabled / visible for fallback camera mouse input.
` / ~	Toggle Debug UI options menu visibility.
Tab	Toggle Settings GUI visibility
F12	Toggle VBS Blue engine **Diag Manager GUI** for advanced un-documented engine diagnostics.
Alt + **Enter**	Toggle full-screen mode on / off.
Alt + **F4**	Closes the VBS Blue IG client

6.3 Fallback Camera Controls

The fallback camera is visible when no other cameras have been created by a simulation host. If a camera has been created by a simulation host, the fallback camera is disabled. It can be re-enabled by enabling **Debug UI / BlueIG Blue Systems / Overrides / Force Fallback Camera**.

You can control the camera using the keyboard and mouse, and also select camera locations for the camera to jump to.

- Camera Movement (below)
- Camera Location Selection (on the next page)

6.3.1 Camera Movement

The keyboard and mouse can be used to control the camera view relative to the Earth.

Press **Esc** to show / hide the cursor. Mouse and keyboard controls are enabled, when the cursor is visible.

Use the following commands to control the camera:

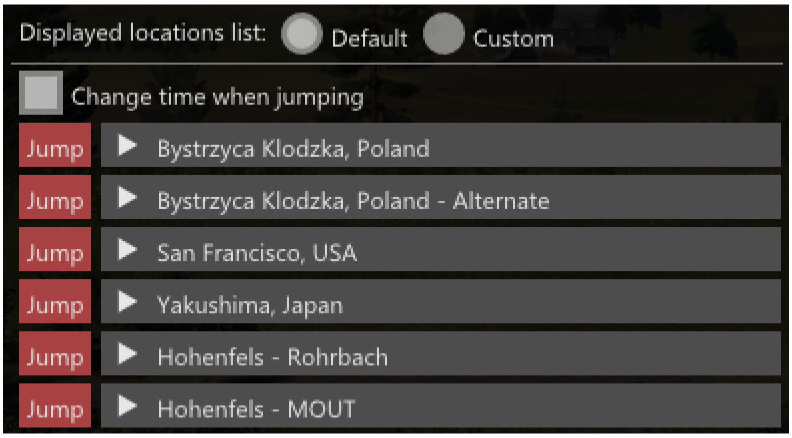
Key	Action
W	Move camera forward along the Earth surface.
S	Move camera backward along the Earth surface.
A	Move camera left along the Earth surface.
D	Move camera right along the Earth surface.
R	Elevate camera away from the Earth center.
F	Lower camera toward the Earth center.
RMB	Move the mouse while holding the RMB to pitch the camera.
Scroll wheel	Increase / decrease camera speed.
Hold LCtrl + W / S / A / D / R / F	Move camera slowly.
Hold LShift + W / S / A / D / R / F	Move camera quickly.
Double-RMB	Move camera to double-clicked location in a straight line.
Double-MMB	Move camera to double-clicked location in a parabolic line.
Ctrl + Alt + S	Saves the current position and camera state to the Windows clipboard, including date and time.
Ctrl + Alt + L	Restores the saved camera position / state.

6.3.2 Camera Location Selection

In addition, you can assign locations for the camera to jump to.

Follow these steps:

1. Press backquote (`) / tilde (~) to access the Debug UI (on page 376), and select **Camera List Window**.
2. In the **Saved Camera Locations** panel, scroll to the camera location that you added, and click **Jump**.



The camera view changes to the selected location.

To delete a camera location using the UI, click **[X]** and then **[Y]** to confirm. Note this is only available when "Custom" is selected, as default locations cannot be deleted.

To add / delete a camera location using XML configuration, see Camera Locations (on page 435).

6.4 VBS Blue IG Settings

Modify VBS Blue IG XML component settings files using the **Settings** graphical interface.

> **NOTE**
>
> Settings can also be adjusted directly by modifying the XML files associated with the setting.
>
> Settings are not designed to be compatible between versions. When updating to new versions of VBS Blue IG, it is recommended that any changes to settings be re-applied using the Settings UI.

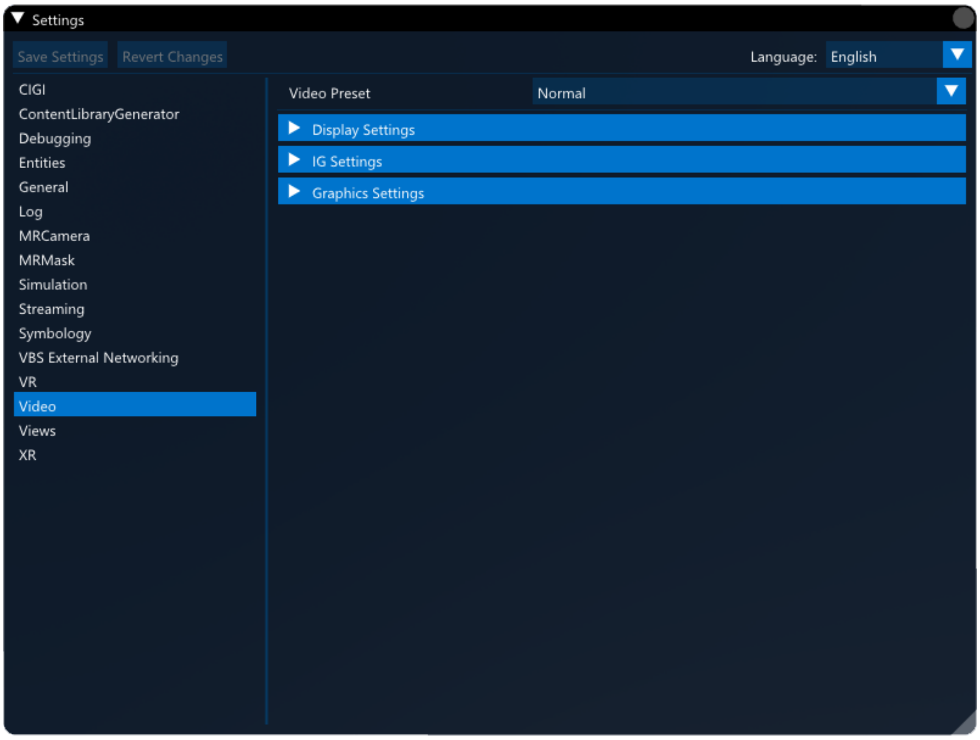
- Basic Usage (below)
- Configuration Options (on the next page)

6.4.1 Basic Usage

Configure VBS Blue IG settings in the Settings window.

Follow these steps:

1. Press **Tab** to open the Settings UI.



2. In the left panel, select a setting to configure or update.

   The configuration settings appear in the right panel.

   For more information about the different settings available, see Configuration Options (below).

3. Update each setting as needed.

   > **TIP**
   >
   > Place the cursor over each setting to display a tool tip that explains its function and purpose.

4. To save the changes, click **Save Settings**.

   > **WARNING**
   >
   > If you want to revert the changes and restore the original values within the XML files, click **Revert Changes** before clicking **Save Settings**. Otherwise, if you click **Save Settings** before **Revert Changes**, the changes cannot be reverted.

The VBS Blue IG configuration XML files are updated.

These XML files are located in:

`%LOCALAPPDATA%\Bohemia Interactive Simulations\VBS Blue IG\`*`version`*`\Settings\`.

> **NOTE**
>
> The Settings folder is a sub-directory in the Product Directory. This can be overridden by the `-productDir` option described in Startup Parameters (on page 454).
>
> If you update the configuration XML files with the Settings UI or manually during runtime, the application must be restarted for the new settings to take effect.

6.4.2 Configuration Options

The left panel of the Settings UI shows the following VBS Blue IG configuration options:

- **CIGI** - CIGI protocol settings for Sessions, Queries and Symbology. For more information, see CIGI Settings (on page 205).
- **Content Library Generator** - Provides the ability to export content library XML files and screenshots for content in VBS Blue IG. For more information, see .
- **Debug UI** - Setting related to the Debug User Interface. For more information, see Debug UI Settings (on page 228).

- **Debugging** - Settings related to debugging and diagnostics. For more information, see Debugging Settings (on page 230).
- **Draw Settings** - Settings for the Draw component. For more information, see Draw Component Settings (on page 237).
- **Entities** - Lifeform and Vehicle settings. For more information, see Entities Settings (on page 238).
- **General** - Task Threading and Headless Mode settings. For more information, see General Settings (on page 255).
- **Log** - Log settings for errors, warnings, and other troubleshooting information. For more information, see Log Settings (on page 257).
- **MRCamera** - Mixed Reality camera settings. For more information, see MRCamera Settings (on page 278).
- **MRMask** - Mixed Reality alpha mask settings. For more information, see MRMasking Component (on page 76) and MRMask Settings (on page 281).
- **Simulation** - Simulation settings, including Laser Configuration (on page 441). For more information, see Simulation Settings (on page 285).
- **Streaming** - Video streaming settings. For more information, see Streaming Settings (on page 314).
- **Symbology** - Symbology settings. For more information, see Symbology Settings (on page 319).
- **VBS External Networking** - VBS external networking settings. For more information, see VBS External Networking Settings (on page 324).
- **VR** - Customize the settings for VR Devices for a simulation host. For more information, see VR Devices Settings (on page 368).
- **Video** - Graphics quality and other related video settings. For more information, see Video Settings (on page 340).
- **Views** - View settings. For more information, see Views Settings (on page 361).
- **XR** - Customize the XR Training Platform settings. For more information, see XR Settings (on page 372).

> **NOTE**
> All the options above can be edited directly within their corresponding XML file, for example, `CIGI.xml`.

Settings Override - Settings override allows each IG to override settings from a shared *base set* of settings. For more information, see Settings Override (on page 449).

For more information on the VBS Blue IG configuration files, see Advanced Configuration (on page 434).

NOTE

Settings Override is not an option available within the Settings UI.

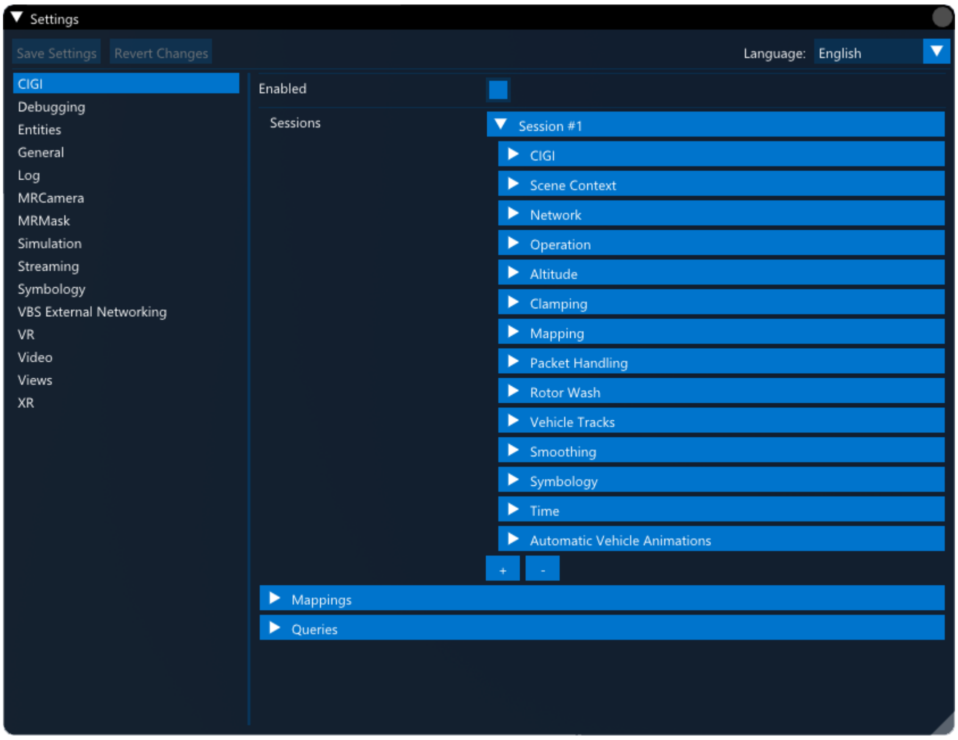
6.4.3 CIGI Settings

Customize CIGI protocol settings for a simulation host using either of the following methods:

- **Settings UI** - Press **Tab** to open the VBS Blue IG Settings (on page 201) and make adjustments, as required.

  In the open panel, select the preferred **Version** in the dropdown menu.

  > **WARNING**
  >
  > Prior versions of VBS Blue IG defaulted to CIGI 3.3.
  >
  > From VBS Blue IG 23.1, the default is **Autodetect**.
  >
  > This may cause issues with CIGI Hosts that do not expect to receive different versions of the CIGI Start of Frame packet.
  >
  > In these cases, select the exact version of CIGI to use.

  > **NOTE**
  >
  > Changing any session settings in the Settings GUI will trigger a CIGI Protocol reset upon saving.

- **Edit XML file** - Open the `CIGI.xml` file, located in the following directory and edit, as required:

  `%LOCALAPPDATA%\Bohemia Interactive Simulations\`
  `VBS Blue IG\`*`version`*`\Settings\`

The tables below describe the type and purpose for each setting.

6.4.3.1 CIGI

Parameter (type)	Values
Enabled **(boolean)** Enable CIGI protocol in VBS Blue IG. If disabled, CIGI will not create any network sockets for processing CIGI, and will therefore be non-functional. .	**Default** true

6.4.3.2 CIGI / Sessions / Session / CIGI

Parameter (type)	Values
Version **(enum)** CIGI version to use on this session.	• 4.0 • 3.3 • 3.2 • 2.0/2.1 • Autodetect **Default** Autodetect

6.4.3.3 CIGI / Sessions / Session / Scene Context

Settings of the session's scene context. A scene context is the unique state of entities, views, etc. independent from that of other scene contexts. A single scene context may be used by multiple sessions, allowing a host to either operate together with, or completely independently from other hosts.

Parameter (type)	Values
Scene Context ID **(uint8)** ID of the scene context the session should use. Multiple sessions may use the same scene context.	**Default** 0

6.4.3.4 CIGI / Sessions / Session / Network / Receive

Settings for the incoming connection.

Parameter (type)	Values
Port **(uint16)** Port to listen on, for example, the port the host sends packets to.	**Min / Max** 1 - UINT16_ MAX **Default** 8003
Multicast Enabled **(boolean)** Whether the network socket should join a multicast group to receive multicast packets. When enabled, #MulticastAddress has to be specified.	**Default** false
Multicast Address **(string)** Multicast group address to join. This option is ignored when #MulticastEnabled is false.	**Default** 225.0.0.1
Interface **(string)** Address of the network adapter to send the packets from. Use 0.0.0.0 to select the primary network adapter.	**Default** 0.0.0.0

6.4.3.5 CIGI / Sessions / Session / Network / Send

Settings for the outgoing connection.

Parameter (type)	Values
Enabled **(boolean)** Enable CIGI messages to be sent to the Host. If disabled, CIGI will only receive messages from the Host, will not send any Start of Frame or other response packets, and queries are implicitly disabled.	**Default** true
Address **(string)** Address of the host. This can be a multicast address.	**Default** 127.0.0.1
Port **(uint16)** Port to send to, i.e. the host's receive port.	**Min / Max** 1 - UINT16_MAX **Default** 8004
Interface **(string)** Address of the network adapter to send the packets from. Use 0.0.0.0 to select the primary network adapter.	**Default** 0.0.0.0
Send From Port **(uint16)** Local port to use for sending. If set to 0 a random available port will be selected.	**Min / Max** 0 - UINT16_MAX **Default** 0
Time To Live (TTL) **(uint8)** For multicast traffic, it might be necessary to override the system default value for Time-to-live / Hop limit IP header field. For example, when the system default for TTL is 1, the multicast traffic is limited to the same subnet. This setting, when non-zero, overrides the system default, thus enabling to control the range the packet is forwarded to. Set to zero to use the system default.	**Min / Max** 0 - UINT8_MAX **Default** 0

6.4.3.6 CIGI / Sessions / Session / Operation / Execution

Parameter (type)	Values
Query Order **(enum)** The order in which queries are performed during the frame.	• Synchronous queries • Async complex queries • Fully async queries **Default** Synchronous queries
Force Settings **(boolean)** Force execution settings for this session. By enabling this setting, execution properties in the *System Session Settings Component Control Packet* are ignored when sent by a host.	**Default** false

6.4.3.7 CIGI / Sessions / Session / Operation / Synchronous

Parameter (type)	Values
Enabled (boolean) Enable CIGI Synchronous operation. If True, the IG will wait for a message from the Host before processing the frame. If False, the IG will always perform in Asynchronous operation, and never wait for the Host to send data before processing the frame.	**Default** false
Max Latency (uint32) If a message has not been received within #MaxLatency milliseconds, the IG will render the frame without receiving from the Host. The Host will be notified of such event in the CIGI Start Of Frame packet - the "Last Host Frame Number" parameter will not have updated. This may be caused by the network dropping the packet, and the Host should re-send the last packet.	**Default** 5
Max Receive Attempts (uint32) If a message has not been received for #MaxReceiveAttempts frames, the IG will fall back to Asynchronous operation until the Host sends new data.	**Default** 3
Force Settings (boolean) Force synchronous settings for this session. By enabling this setting, synchronous properties in the *System Session Settings Component Control Packet* are ignored when sent by a host.	**Default** false

6.4.3.8 CIGI / Sessions / Session / Operation / Verification

Parameter (type)	Values
Host Frame Number (boolean) Enable verification of frame numbers received in an IG Control packet. If a frame number increments by more than 1, a warning message will be printed to the log, which may be the result of a UDP message being dropped. Disable if the host does not increment the frame number for each IG Control packet sent to the IG.	**Default** true
Force Settings (boolean) Force automatic vehicle animation settings for this session. By enabling this setting, properties in the System Global Simulated Animations Component Control Packet are ignored when sent by a host.	**Default** false

6.4.3.9 CIGI / Sessions / Session / Altitude

Parameter (type)	Values
Use Geoid **(boolean)** CIGI specification defines "Altitude = 0" = "WGS84 Ellipsoid Surface" = "Mean Sea Level" for simplicity reasons. However, this is an inaccurate assumption, as "WGS84 Ellipsoid Surface" does NOT equal "Mean Sea Level", which is affected by the Geoid, and is different throughout the world. If True, Altitude = 0 corresponds to Mean Sea Level. If False, Altitude = 0 corresponds to WGS84 Ellipsoid Surface.	**Default** true
Force Settings **(boolean)** Force altitude settings for this session. By enabling this setting, altitude properties in the *System Session Settings Component Control Packet* are ignored when sent by a host.	**Default** false

6.4.3.10 CIGI / Sessions / Session / Clamping

Parameter (type)	Values
Ignore Altitude Offset **(boolean)** Enable to ignore the Altitude parameter for an entity, when Ground Clamp = Non-Conformal, or Conformal. Disable to use CIGI compliant behavior, where Altitude is relative to terrain or sea level.	**Default** false
Ignore Orientation Offsets **(boolean)** Enable to ignore the Pitch and Roll parameters for an entity, when Ground Clamp = Conformal. Disable to use CIGI compliant behavior, where Pitch and Roll is relative to terrain or sea slope.	**Default** false
Include Objects **(boolean)** Enable to include terrain object surfaces (e.g. buildings) during ground clamping for an entity when Ground Clamp = Non-Conformal, or Conformal. Disable to use CIGI compliant behavior, where object surfaces are ignored.	**Default** false
Include Simulation **(boolean)** Enable to include simulated surfaces (e.g. water waves) during ground clamping for an entity when Ground Clamp = Non-Conformal, or Conformal.	**Default** true

6.4.3.11 CIGI / Sessions / Session / Environment

Parameter (type)	Values
Cumulus Cloud Type (float32) The value to use for the cloud type parameter, when using a CIGI Weather Control packet to set the cloud type to Cumulus. 0 represents a fully cumulus cloud, and 1 represents a fully stratus cloud.	**Min / Max** 0.0000 - 1.0000 **Default** 0.18000
Stratus Cloud Type (float32) The value to use for the cloud type parameter, when using a CIGI Weather Control packet to set the cloud type to Stratus. 0 represents a fully cumulus cloud, and 1 represents a fully stratus cloud.	**Min / Max** 0.0000 - 1.0000 **Default** 0.87000

6.4.3.12 CIGI / Sessions / Session / Mapping

Parameter (type)	Values
Override Material (boolean) Override Material ID with custom mapping defined in Materials.mapping.	**Default** false

6.4.3.13 CIGI / Sessions / Session / Packet Handling

Parameter (type)	Values
Enable Component Control Packets (boolean) Enable processing of the Component Control, or Short Component Control packets. If disabled, these packets will be ignored, and not processed.	**Default** true
Enable Extension Packets (boolean) Enable processing of non-standard CIGI Extension data packets. If disabled, all extension packets will be ignored, and not processed.	**Default** true
Enable Query Packets (boolean) Enable / Disable CIGI query messages (i.e. HAT/HOT, LOS, etc.). If True, enables requests from CIGI Hosts. Potential for performance hits when handling requests. If False, disables all request functions from a CIGI Host. This also means no responses are sent from this IG.	**Default** true
Enable Symbology Packets (boolean) Enable processing of symbol packets. If disabled, these packets will be ignored, and not processed.	**Default** true

6.4.3.14 CIGI / Sessions / Session / Rotor Wash

Parameter (type)	Values
Enable Rotor Wash Effects By Default (boolean) When enabled, helicopter entities will automatically generate rotor wash effects when near the terrain, provided the main rotor is spinning at sufficient RPM. To set this, use the CIGI Rate Control or Velocity Control packet to set an appropriate Yaw Angular Velocity to a Part ID corresponding to a part contained within the 'rotorh' animation source. The specific Part ID to use differs based on the entity type, and can be determined using information in the Content Library. When disabled, rotor wash has to be enabled manually, either for the session using the System Session Settings Component Control packet, or for individual entities using the Entity Vehicle State Component Control packet.	**Default** false

6.4.3.15 CIGI / Sessions / Session / Vehicle Tracks

Parameter (type)	Values
Tracks State **(enum)** Enable state and behavior of tracks generation for vehicles.	• Use Global Setting • Auto • Forced Off **Default** Use Global Setting
Dust Trails State **(enum)** Enable state and behavior of dust trails generation for vehicles.	• Use Global Setting • Auto • Forced Off **Default** Use Global Setting

6.4.3.16 CIGI / Sessions / Session / Lifeform Tracks

Parameter (type)	Values
Tracks State **(enum)** Enable state and behavior of tracks generation for lifeforms.	• Use Global Setting • Auto • Forced Off **Default** Use Global Setting
Snow Trails State **(enum)** Enable state and behavior of snow trails generation for lifeforms.	• Use Global Settings • Auto • Forced off **Default** Use Global Settings

6.4.3.17 CIGI / Sessions / Session / Smoothing

Parameter (type)	Values
Enable Extrapolation **(boolean)** When entity smoothing / extrapolation / interpolation is enabled, enabling this setting will configure the entity to support extrapolation. Disabling this setting will prevent the entity from being extrapolated, which will also prevent rate / velocity control from moving the entity.	**Default** true
Enable Interpolation **(boolean)** When entity smoothing / extrapolation / interpolation is enabled, enabling this setting will configure the entity to support interpolation. Disabling this setting will prevent the entity from being interpolated.	**Default** false
Force Smoothing **(enum)** Forcibly enables Linear Extrapolation / Interpolation for all entities that are created or updated via an Entity Control packet received on this session.	**Default** false

6.4.3.18 CIGI / Sessions / Session / Symbology

Parameter (type)	Values
Allow No Perspective Growth On Non Billboard (boolean) CIGI specification defines that the "Perspective Growth Enable" parameter is ignored when the "Billboard" parameter is set to 0 (Non-Billboard). This setting allows overriding this, allowing a host to send "Perspective Growth Enable = 0 (Disabled)" even when the surface is not a billboard.	**Default** false
Show Attached Surfaces To Any View (boolean) By default (True), any entity attached surface will show to any view attached to any entity. Setting this to False will only show entity attached surfaces on views that are attached to the same entity.	**Default** true

6.4.3.19 CIGI / Sessions / Session / Time

Parameter (type)	Values
UTC Time Offset (int32) UTC hour offset when time is defined as local time (CIGI version <= 3.2).	**Default** 0
Use Host Timestamps (boolean) Enable processing of the Timestamp / Timing Value parameter in IG Control packets. If disabled, the parameter will be ignored, and time will be controlled by the IG.	**Default** true
Timestamp Fallback Time (uint32) When Use Host Timestamps is enabled, the maximum time in milliseconds the IG will wait for the host to send an IG Control packet with a valid Timestamp before falling back to automatically progressing the simulation time. When set to 0, the IG will always rely on the host to provide the simulation time. Note that this will result in the simulation time not progressing when no host is connected.	**Default** 2000
Time Scale (float32) Frame timing value scale for CIGI Hosts expecting non-compliant time scales.	**Min / Max** 0.001 - 1000000 **Default** 1.0000

6.4.3.20 CIGI / Sessions / Session / Automatic Animations

Parameter (type)	Values
Wheel and Track Rotation **(enum)** Controls whether vehicles have automatic wheel and track rotation animations enabled by default. The automatic rotation is simulated based on the vehicle's forward velocity. Disable if such animations are not required, or manual control over animations is desired.	• Use Global Setting • Enabled • Disabled **Default** Use Global Setting
Steering **(enum)** Controls whether vehicles have automatic wheel steering animations enabled by default. The automatic steering is simulated based on the vehicles yaw angular velocity. Disable if such animations are not required, or manual control over animations is desired.	• Use Global Setting • Enabled • Disabled **Default** Use Global Setting
Suspension **(enum)** Controls whether vehicles have automatic suspension animations enabled by default. The automatic suspension is simulated based on the vehicles height above the ground. Disable if such animations are not required, or manual control over animations is desired.	• Use Global Setting • Enabled • Disabled **Default** Use Global Setting
Time **(enum)** Controls whether time-based animations are automatically simulated by default. When enabled, entity animations controlled by the simulation and clock time will be automatically updated. Disable if such animations are not required, or manual control over animations is desired.	• Use Global Setting • Enabled • Disabled **Default** Use Global Setting

6.4.3.21 CIGI / Mappings

Parameter (type)	Values

Mappings Directories (string)
A list of directories that CIGI mappings will be loaded from. Mappings are loaded in the order in which they are specified, with any conflicting or duplicate mappings overwriting the previously defined mappings. These directories can be specified as either a relative path to the VBS Blue IG installation directory or an absolute path.
	Default directory
	`\`*`IG_Installation`*`\data\`
	`\BlueProduct\mappings\Components\`
	`CigiProtocol\Mappings\`

6.4.3.22 CIGI / Queries / HAT/HOT

Parameter (type)	Values
Min Detail **(uint8)** Specifies the minimum detail of the HAT/HOT request. If the area has not yet loaded, the IG will hang until the minimum detail is loaded. Value represents LOD, with 0 being lowest quality, and 23 being the highest quality. Specifying 0 will use the global MinDetail settings from BlueController.	**Min / Max** 0 - 23 **Default** 0
Max Detail **(uint8)** Specifies the maximum detail of the HAT/HOT request. Value represents LOD, with 0 being lowest quality, and 23 being the highest quality. Specifying 0 will use the global MaxDetail settings from BlueController.	**Min / Max** 0 - 23 **Default** 0
Required Detail **(uint8)** Specifies the detail the HAT/HOT result needs to be greater than or equal to, in order to process the result and send the HAT/HOT Response packet to the Host. If the result is of less detail than specified (i.e. the area has not yet loaded), then the result will not be processed, and it will execute again on the next frame, repeatedly until it returns higher detail. Value represents LOD, with 0 being lowest quality, and 23 being the highest quality. Specifying 0 will skip any possible delay in results.	**Min / Max** 0 - 23 **Default** 0
Include Simulation **(boolean)** Specifies if HAT/HOT recognizes dynamic surfaces (such as water waves).	**Default** true
Enable Surface Normal Query **(boolean)** Specifies if HAT/HOT calculates the surface normal or not. Can be disabled to improve query performance for extended responses, if the surface normal is not required.	**Default** true
Enable Surface Material Query **(boolean)** Specifies if HAT/HOT calculates the surface material or not. Can be disabled to improve query performance for extended responses, if the surface material is not required.	**Default** true
Surface Request Types **(enum)** Specifies the type of surface query the HAT/HOT calculates. May support multiple values: Ground, Water.	• Ground • Water • Objects **Default** Ground Water

6.4.3.23 CIGI / Queries / Line Of Sight

Parameter (type)	Values
Min Detail **(uint8)** Specifies the minimum detail of the Line Of Sight request. If the area has not yet loaded, the IG will hang until the minimum detail is loaded. Value represents LOD, with 0 being lowest quality, and 23 being the highest quality. Specifying 0 will use the global MinDetail settings from BlueController.	**Min / Max** 0 - 23 **Default** 0
Max Detail **(uint8)** Specifies the maximum detail of the Line Of Sight request. Value represents LOD, with 0 being lowest quality, and 23 being the highest quality. Specifying 0 will use the global MaxDetail settings from BlueController.	**Min / Max** 0 - 23 **Default** 0
Required Detail **(uint8)** Specifies the detail the Line Of Sight result needs to be greater than or equal to, in order to process the result and send the Line Of Sight Response packet to the Host. If the result is of less detail than specified (i.e. the area has not yet loaded), then the result will not be processed, and it will execute again on the next frame, repeatedly until it returns higher detail. Value represents LOD, with 0 being lowest quality, and 23 being the highest quality. Specifying 0 will skip any possible delay in results.	**Min / Max** 0 - 23 **Default** 0
Max Intersections **(uint8)** Maximum number of intersections to return when performing a Line of Sight Segment Request, or Line of Sight Vector Request.	**Min / Max** 0 - UINT8_MAX **Default** 10
Enable Materials **(boolean)** When enabled, Line of Sight queries will perform material queries against intersections, and return material codes in response packets. When disabled, response packets will not contain valid material codes, however performance will be slightly increased. .	**Default** true
Geometry Type **(enum)** Defines the selected object geometry type to perform intersections against. Different geometry types have different levels of fidelity, and Ballistic geometry is the only type that contains materials. This setting only applies to object intersections, it is ignored for other intersection types (such as ground, water, biome, etc.).	• Collision • View • Ballistic **Default** Ballistic

6.4.3.24 CIGI / Queries / Line Of Sight / Segment

Parameter (type)	Values
Source Entity Ignore Type (enum) When performing a Line Of Sight Segment Request, allow for ignoring of intersections with entities related to the source entity. Multiple options can be selected at the same time.	• No Ignore • Ignore Entity • Ignore Ancestors • Ignore Descendants **Default** No Ignore
Destination Entity Ignore Type (enum) When performing a Line Of Sight Segment Request, allow for ignoring of intersections with entities related to the source entity. Multiple options can be selected at the same time.	• No Ignore • Ignore Entity • Ignore Ancestors • Ignore Descendants **Default** No Ignore

6.4.3.25 CIGI / Queries / Line Of Sight / Vector

Parameter (type)	Values
Source Entity Ignore Type (enum) When performing a Line Of Sight Vector Request, allow for ignoring of intersections with entities related to the source entity. Multiple options can be selected at the same time.	• No Ignore • Ignore Entity • Ignore Ancestors • Ignore Descendants **Default** NoIgnore

6.4.3.26 CIGI / Queries / Collision Detection Segment

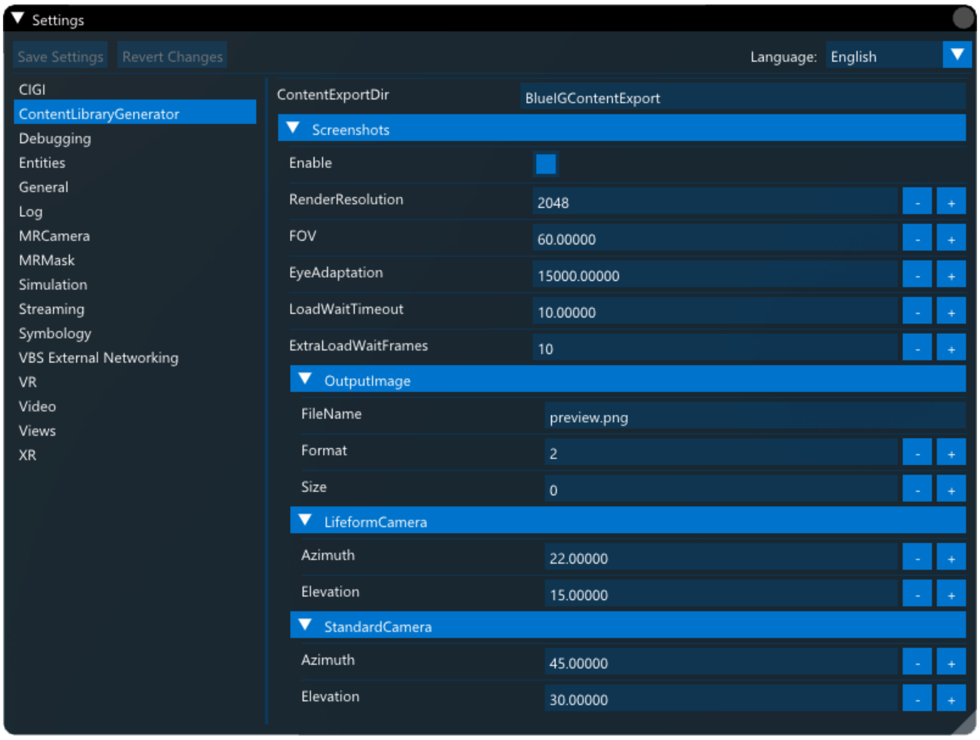
Parameter (type)	Values
Min Detail **(uint8)** Specifies the minimum detail of the Collision Detection Segment request. If the area has not yet loaded, the IG will hang until the minimum detail is loaded. Value represents LOD, with 0 being lowest quality, and 23 being the highest quality. Specifying 0 will use the global MinDetail settings from BlueController.	**Min / Max** 0 - 23 **Default** 0
Max Detail **(uint8)** Specifies the maximum detail of the Collision Detection Segment request. Value represents LOD, with 0 being lowest quality, and 23 being the highest quality. Specifying 0 will use the global MaxDetail settings from BlueController.	**Min / Max** 0 - 23 **Default** 0
Required Detail **(uint8)** Specifies the detail the Collision Detection Segment result needs to be greater than or equal to, in order to process the result and send the Collision Detection Segment Notification packet to the Host. If the result is of less detail than specified (i.e. the area has not yet loaded), then the result will not be processed, and it will execute again on the next frame, repeatedly until it returns higher detail. Value represents LOD, with 0 being lowest quality, and 23 being the highest quality. Specifying 0 will skip any possible delay in results.	**Min / Max** 0 - 23 **Default** 0
Entity Ignore Type **(enum)** When performing a Collision Detection Segment Request, allow for ignoring of intersections with entities related to the source entity. Multiple options can be selected at the same time.	• NoIgnore • IgnoreEntity • IgnoreAncestors • IgnoreDescendants **Default** IgnoreEntity
Enable Materials **(boolean)** When enabled, Collision Detection Segment queries will perform material queries against intersections, and return material codes in response packets. When disabled, response packets will not contain valid material codes, however performance will be slightly increased. .	**Default** true

Parameter (type)	Values
Geometry Type (enum) Defines the selected object geometry type to perform intersections against. Different geometry types have different levels of fidelity, and Ballistic geometry is the only type that contains materials. This setting only applies to object intersections, it is ignored for other intersection types (such as ground, water, biome, etc.). .	• Collision • View • Ballistic **Default** Ballistic

6.4.4 Content Library Generator

Customize the Content Library Generator settings using either of the following methods:

- **Settings UI** - Press **Tab** to open the VBS Blue IG Settings (on page 201) and make adjustments, as required.



- **Edit XML file** - Open the `ContentLibraryGenerator.xml` file located in the following directory and edit, as required:

  `%LOCALAPPDATA%\Bohemia Interactive Simulations\VBS Blue IG\`*`version`*`\Settings\`

Additional functionality is available in the Debug UI Content Library Generator (on page 388).

The table below describes the type and purpose for each setting.

6.4.4.1 ContentLibraryGenerator

Parameter (type)	Values
ContentExportDir (string) Path of the directory into which the content configuration is exported. Can be absolute or relative to the component directory.	**Default** ContentExport

6.4.4.2 ContentLibraryGenerator / Screenshots

Parameter (type)	Values
Enable **(boolean)** Whether screenshots of the content should be captured. When **false**, the rest of the parameters in this section are ignored.	**Default** True
RenderResolution **(int32)** Whether screenshots of the content should be captured. When `false`, the rest of the parameters in this section are ignored.	**Default** 2048
FOV **(float64)** Horizontal and vertical field of view of the camera used to take the screenshot.	**Default** 60.00000
EyeAdaptation **(float64)** Fixed eye adaptation value to use when taking the screenshots.	**Default** 15000.00
LoadWaitTimeout **(float64)** Maximum time in seconds to wait for an object to be fully loaded before timing out.	**Default** 10.00000
ExtraLoadWaitFrames **(int32)** Number of frames to wait once an object reports being fully loaded before taking the screenshot. This may be necessary to ensure everything, including animations, had been fully initialized.	**Default** 10

6.4.4.3 ContentLibraryGenerator / Screenshots / OutputImage

Parameter (type)	Values
FileName **(string)** Name of the export image, including the extension.	**Default** preview.png
Format **(int32)** Format of the output image as ScreenshotImageFormat_v2 enum value	**Default** 2
Size **(int32)** Width and height of the image in pixels. When 0, the width and height of the rendered viewport are used.	**Default** 0

6.4.4.4 ContentLibraryGenerator / Screenshots / LifeformCamera

Parameter (type)	Values
Azimuth **(float64)** Azimuth of the camera in degrees relative to lifeforms when taking their screenshot. When 0, the camera will be positioned forward of the object.	**Default** 22.00000
Elevation **(float64)** Elevation of the camera in degrees relative to lifeforms when taking their screenshot. When 0, the camera is positioned level with the object.	**Default** 15.00000

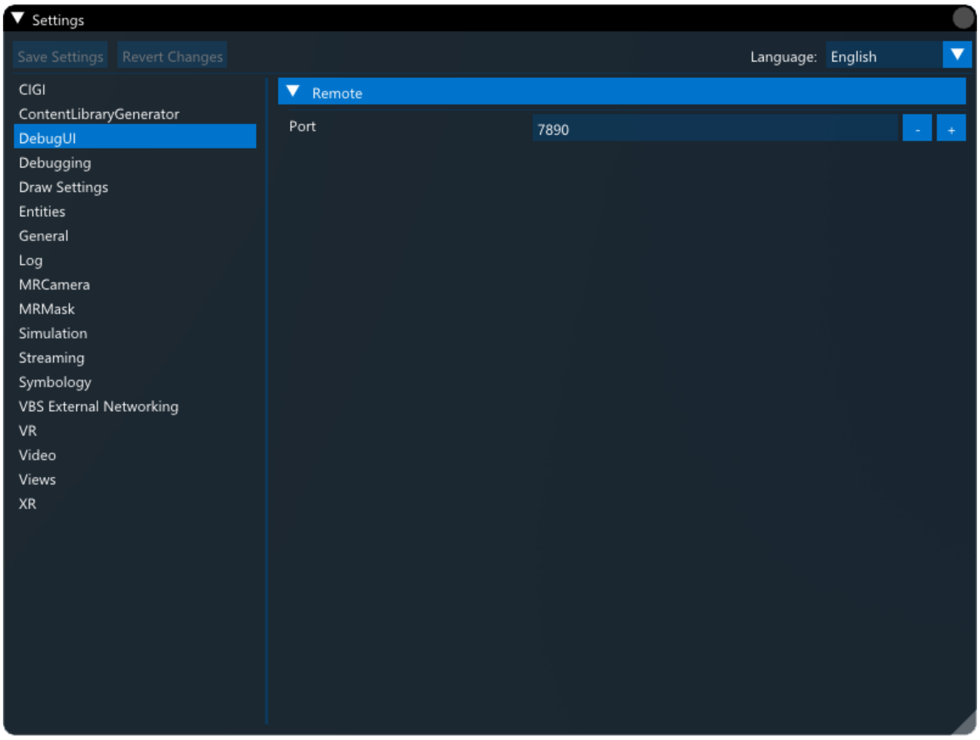
6.4.4.5 ContentLibraryGenerator / Screenshots / StandardCamera

Parameter (type)	Values
Azimuth **(float64)** Azimuth of the camera in degrees relative to all other object types when taking their screenshot. When 0, the camera is positioned forward of the object.	**Default** 45.00000
Elevation **(float64)** Elevation of the camera in degrees relative to all other object types when taking their screenshot. When 0, the camera is positioned level with the object.	**Default** 30.00000

6.4.5 Debug UI Settings

Customize the Debug UI settings using either of the following methods:

- **Settings UI** - Press **Tab** to open the VBS Blue IG Settings (on page 201) and make adjustments, as required.



- **Edit XML file** - Open the `DebugUI.xml` file located in the following directory and edit, as required:

  `%LOCALAPPDATA%\Bohemia Interactive Simulations\VBS Blue IG\` *`version`*`\Settings\`

The table below describes the type and purpose for each setting.

6.4.5.1 DebugUI / UI Scale

Parameter (type)	Values
AutoScale **(boolean)** If true, the UI is automatically scaled based on the screen size.	**Default** false
FixedScale **(float32)** Fixed scale to use for the UI. This value is ignored if AutoScale == true.	**Min / Max:** 1.000000 - 20.000000 **Default** 1.000000

6.4.5.2 DebugUI / UI Scale / DefaultResolution

Parameter (type)	Values
Width **(unint32)** The native resolution used for the UI. The scale at the default resolution always equals one.	**Min / Max:** 1 - 0 **Default:** 1920
Height **(unint32)** The native resolution used for the UI. The scale at the default resolution always equals one.	**Min / Max:** 1 - 0 **Default:** 1080

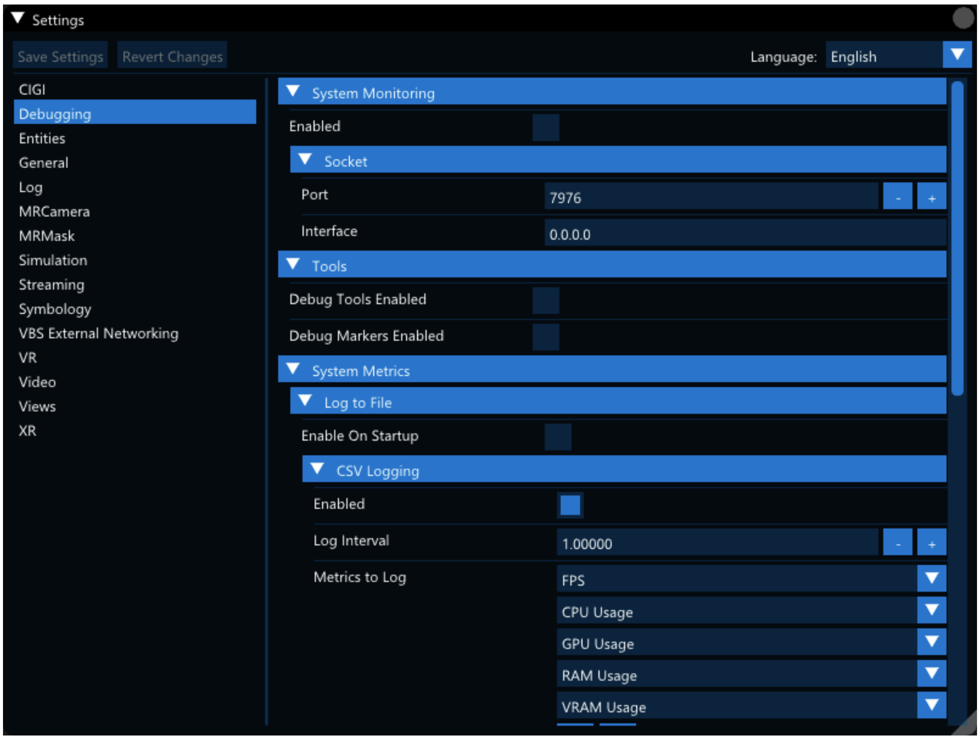
6.4.5.3 DebugUI / Remote

Parameter (type)	Values
Port **(uint16)** Port to access Debug UI via web browser. Use **0** to disable.	**Min / Max:** 1 - 0 **Default** 7890

6.4.6 Debugging Settings

Customize **System Monitoring**, **System Metrics** and other debug **Tools** using either of the following methods:

- **Settings UI** - Press **Tab** to open the VBS Blue IG Settings (on page 201) and make adjustments, as required.



  Additional runtime-only debug settings can be enabled within the Debug UI. For more information, see Debug UI (on page 376).

- **Edit XML file** - Open the `Debugging.xml` file, located in the following directory and edit, as required:

  `%LOCALAPPDATA%\Bohemia Interactive Simulations\VBS Blue IG\version\Settings\`

> **NOTE**
>
> If metrics logging to file is enabled on startup, the created CSV and / or JSON files are output into the following directory:
>
> `%LOCALAPPDATA%\Bohemia Interactive Simulations\VBS Blue IG\version\PerformanceLogs\`
>
> For more information, see Debugging / System Metrics / Log To File (on page 233).

The following tables describe the type and purpose for each setting.

6.4.6.1 Debugging / CrashReporter / CrashDump

Parameter (type)	Values
DumpOutputLocation (string)	**Default** %LOCALAPPDATA%\Bohemia Interactive Simulations\VBS Blue IG*version*\CrashDumps\
DumpName (string)	**Default** CrashDump
DumpType (unint32)	**Default** 266565

6.4.6.2 Debugging / CrashReporter / CrashTypeFlags

Parameter (type)	Values
SEHException (boolean) SEHException	**Default** true
Terminate (boolean) Terminate	**Default** true
Unexpected (boolean) Unexpected	**Default** true
Unexpected (boolean) Unexpected	**Default** true
InvalidParameter (boolean) InvalidParameter	**Default** true
NewOperatorFault (boolean) NewOperatorFault	**Default** true
SIGABRT (boolean) SIGABRT	**Default** true
SIGFPE (boolean) SIGFPE	**Default** true
SIGILL (boolean) SIGILL	**Default** true
SIGINT (boolean) SIGINT	**Default** true
SIGSEGV (boolean) SIGSEGV	**Default** true
SIGTERM (boolean) SIGTERM	**Default** true

6.4.6.3 Debugging / System Monitoring

Parameter (type)	Values
Enabled **(boolean)** Enables or disables the system monitoring network.	**Default** false

6.4.6.4 Debugging / System Monitoring / Socket

Parameter (type)	Values
Port **(uint16)** The TCP port to bind to for the system monitoring network	**Default** 7976
Interface **(string)** Address of the network adapter to bind on for the system monitoring network. Use 0.0.0.0 to bind to the default network adapter.	**Default** 0.0.0.0

6.4.6.5 Debugging / Tools

Parameter (type)	Values
Debug Tools Enabled **(boolean)** If True, extra debug tools will be enabled in VBS Blue IG.	**Default** false
Debug Markers Enabled **(boolean)** If True, various visual debug markers will be enabled in the scene.	**Default** false

6.4.6.6 Debugging / System Metrics / Log To File

Parameter (type)	Values
Enable On Startup **(boolean)** Enable logging performance metrics to file on application launch.	**Default** false

6.4.6.7 Debugging / System Metrics / Log to File / CSV Logging

Parameter (type)	Values
Enabled **(boolean)** Controls whether logging of this output file type is enabled.	**Default** false
Log Interval **(float32)** Controls the time interval to log min, max, and average values for all metrics.	**Min / Max** 10000 - 0.00001 **Default** 1.0000
Metrics to Log **(enum)** Controls which of the following metrics are logged to file: • FPS • CPU Usage • GPU Usage • RAM Usage • VRAM Usage	**Default** • FPS • CPU Usage • GPU Usage • RAM Usage • VRAM Usage

6.4.6.8 Debugging / System Metrics / Log to File / JSON Logging

Parameter (type)	Values
Enabled (boolean)	Controls whether logging of this output file type is enabled. **Default** true
Metrics to Log (enum)	Controls which of the Available System Metrics (below) are logged to file. **Default:** • frameTimeApplication • frameTimeEngine • frameTimeTotal • numBiomeTriangles • numDrawnParticles • numQueries • numSimulatedParticles • numTriangles • sceneLoadTotal • cpuUsageApplication • gpuDataUtilizationApplication • gpuDataVramUsageDxgi • gpuDataVramUsageApplication • gpuRenderingUtilizationApplication • gpuRenderingVramUsageDxgi • gpuRenderingVramUsageApplication • memoryUsageApplicationVirtual

Available System Metrics

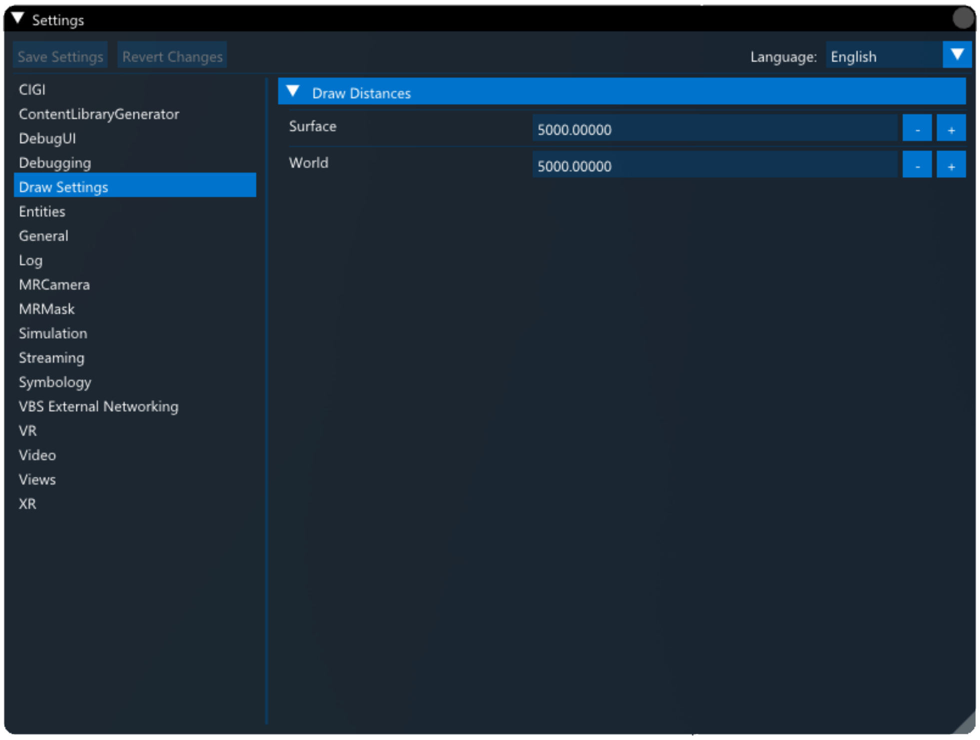
- cpuUsageApplication
- cpuUsageTotal
- frameTimeApplication
- frameTimeEngine
- frameTimeTotal
- memoryUsageApplicationPhysical
- memoryUsageApplicationVirtual
- memoryUsageTotalPhysical
- memoryUsageTotalVirtual
- numBiomeObjects
- numBiomeTriangles
- numDrawnParticles
- numObjects
- numPrimitives
- numQueries
- numSimulatedParticles
- numTriangles
- gpuDataUtilizationApplication
- gpuDataUtilizationTotal
- gpuDataVramUsageApplication
- gpuDataVramUsageDxgi
- gpuRenderingUtilizationApplication
- gpuRenderingUtilizationTotal
- gpuRenderingVramUsageApplication
- gpuRenderingVramUsageDxgi
- gpuRenderingVramUsageTotal
- numGetIntersections
- numGetAirTemperatureBestDataQueries
- numGetAirTemperatureColumnBestDataQueries
- numGetAirTemperatureColumnQueries
- numGetAirTemperatureQueries
- numGetHeightAboveTerrainQueries
- numGetTerrainNormalQueries
- numGetTerrainHeightAboveEllipsoidQueries

- sceneLoadBiome
- sceneLoadClouds
- sceneLoadGeometry
- sceneLoadGround
- sceneLoadObjects
- sceneLoadPointCloud
- sceneLoadTotal
- sceneLoadWater
- numGetTerrainSurfacePositionQueries
- numGetVolumeCollisions
- numGetWindConditionsBestDataQueries
- numGetWindConditionsColumnBestDataQueries
- numGetWindConditionsColumnQueries
- numGetWindConditionsQueries
- numSampleHeightAboveTerrainQueries
- numSampleTerrainHeightAboveEllipsoidQueries
- numSampleTerrainNormalQueries
- numSampleTerrainSurfacePositionQueries

6.4.7 Draw Component Settings

Customize the Draw Component settings using either of the following methods:

- **Settings UI** - Press **Tab** to open the VBS Blue IG Settings (on page 201) and make adjustments, as required.



- **Edit XML file** - Open the `DrawSettings.xml` file located in the following directory and edit, as required:

  `%LOCALAPPDATA%\Bohemia Interactive Simulations\VBS Blue IG\`*`version`*`\Settings\`

The table below describes the type and purpose for each setting.

6.4.7.1 Draw Settings / Draw Distance

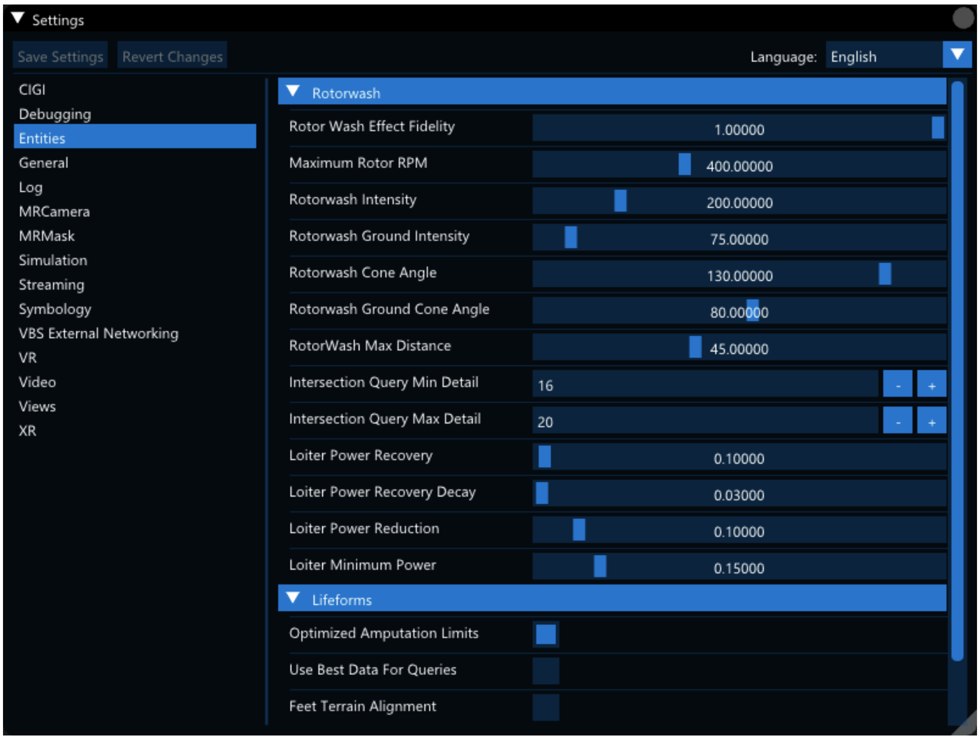
Settings for the Draw component.

Parameter (type)	Values
Surface **(float32)** Draw distance for the surface draw API.	**Default** 5000.00
World **(float32)** Draw distance for the world draw API.	**Default** 5000.00

6.4.8 Entities Settings

Customize Entity settings for a simulation host using either of the following methods:

- **Settings UI** - Press **Tab** to open the VBS Blue IG Settings (on page 201) and make adjustments, as required.



- **Edit XML file** - Open the `Entities.xml` file, located in the following directory and edit, as required:

  `%LOCALAPPDATA%\Bohemia Interactive Simulations\VBS Blue IG\`*`version`*`\Settings\`

The tables below describe the type and purpose for each setting.

6.4.8.1 Entities / Rotorwash

Parameter (type)	Values
Rotor Wash Effect Fidelity (float32) Scales the overall strength of rotor wash effects.	**Min / Max** 0 - 1.00000 **Default** 7.000
Maximum Rotor RPM (float32) RPM value at which the strongest rotor wash effect is reached.	**Min / Max** 60 - 1000 **Default** 400.00
Rotorwash Intensity (float32) Base Intensity of the wind originating from the vehicle.	**Min / Max** 0 - 1000 **Default** 200.00
Rotorwash Ground Intensity (float32) Base Intensity of the wind originating from the ground when the rotorwash collides with the terrain.	**Min / Max** 0 - 1000 **Default** 75.00
Rotorwash Cone Angle (float32) Angle of the wind cone originating from the vehicle.	**Min / Max** 0 - 150 **Default** 130.00
Rotorwash Ground Cone Angle (float32) Angle of the wind cones originating from the collision point of the rotor wash and the terrain.	**Min / Max** 0 - 150 **Default** 80.00
RotorWash Max Distance (float32) Distance from the helicopter and the terrain at which rotor wash should start affecting the ground.	**Min / Max** 10 - 100 **Default** 45.00
Intersection Query Min Detail float32) Specifies the minimum required detail when performing queries.	**Min / Max** 0 - 23 **Default** 0.00

Parameter (type)	Values
Intersection Query Max Detail (float32) Specifies the maximum required detail when performing queries.	**Min / Max** 0 - 23 **Default** 16.00
Loiter Power Recovery (float32) The initial rate rotor wash power recovers when moving after losing power from loitering.	**Min / Max** 0.01 - 10.000 **Default** 0.1000
Loiter Power Recovery Decay (float32) The rate the rotor wash power recovery slows down after the initial boost from moving from loitering.	**Min / Max** 0.01 - 10.000 **Default** 0.0300
Loiter Power Reduction (float32) The rate rotor wash loses its power when loitering over an area for long periods of time.	**Min / Max** 0.01 - 0.95 **Default** 0.1000
Loiter Minimum Power (float32) The minimum rotor wash power that is required for loitering to have any effect.	**Min / Max** 0.01000 - 0.95000 **Default** 0.1500

6.4.8.2 Entities / Lifeforms

Parameter (type)	Values
Optimized Amputation Limits **(boolean)** Hides all lifeform selections with "stub_" prefix in their name. Disable to revert to legacy behavior.	**Default** true
Use Best Data For Queries **(boolean)** Flag to determine if best data should be used for terrain queries. **True** - Use best data. More performance intensive and may cause pauses while best data is loaded. **False** - Do not use best data. Lower fidelity but should not cause performance degradation.	**Default** false
Feet Terrain Alignment **(boolean)** Flag to determine if feet should be aligned to the terrain. **True** - Feet will be aligned to the terrain. **False** - Feet will not be aligned to the terrain.	**Default** false
Animation LOD Fidelity **(float32)** Larger numbers increase fidelity. Animation LOD changes how frequently objects do animation updates. Disabled if fidelity is set to '0', causing characters to update every frame. Each LOD level is [15 * Fidelity * 1.333^LOD] meters. Example: Fidelity=2' make characters within 30m of a camera to be at animation LOD '0', 30-70m at LOD '1'. Each animation LOD will skip 2^LOD-1 frames before animating. LOD '3' will skip 2^3-1 frames. LOD '1' will skip 2^1-1 frames. LOD '0' will 2^0-1 frames.	**Min / Max** 0 - FLOAT_MAX **Default** 1.000

6.4.8.3 Entities / Lifeforms / Tracks

<table>
<tr><th>Parameter (type)</th><th>Values</th></tr>
<tr><td>Enabled
(boolean)
Determines whether the lifeform tracks drawing system is enabled.</td><td>Default
false</td></tr>
<tr><td>Default Track Enable State
(enum)
Default enable state of tracks generation for newly registered lifeforms.

NOTE
Changing Default Track Enable State only applies to newly created units.

Available options:
0: Forced Off - Tracks are not drawn. Use if manual control over track drawing is desired.
1: Auto - Tracks are drawn when conditions defined by the product settings are satisfied.</td><td>Auto
Forced Off
Default
Forced Off</td></tr>
<tr><td>Maximum Track Segments Count
(uint8)
Maximum number of track segments existing at the same time for a (single leg of a) single entity.
When the maximum number of segments is reached, the oldest segment is destroyed and a new segment starts being populated with new footmarks.</td><td>Min / Max
1 - UINT8_MAX
Default
10</td></tr>
<tr><td>Maximum Track Segment Size
(uint8)
Maximum number of footmarks inside a track segment.
When the maximum number of footmarks in a segment is reached, the segment is terminated and a new segment starts being populated with new footmarks.</td><td>Min / Max
1 - UINT8_MAX
Default
6</td></tr>
</table>

6.4.8.4 Entities / Lifeforms / Tracks / Auto Behavior

Parameter (type)	Values
Check Surface Material (boolean) When enabled, material ID of the surface will be checked to see if it allows tracks. When disabled, tracks will be created on all surfaces.	**Default** true
Surface Contact Distance Limit (float32) Limit for distance between a lifeform in the **auto** state and the surface below it in meters for them to be considered to be in contact. **NOTE** Use 0.0 to disable checking for the contact with the surface.	**Min / Max** 0 - 1000 **Default** 0.5000

6.4.8.5 Entities / Lifeforms / Snow Trails

Parameter (type)	Values
Snow Trails Enabled (boolean) Determines whether the snow trails generation system is enabled.	**Default** false
Default Snow Trails Enable State (enum) Default enable state of snow trails generation for newly registered lifeforms. Available options: • 0: Forced Off - The snow trail source is forced to be off. • 1: Auto - Snow trails are generated when conditions defined by the product settings are satisfied.	• Auto • Forced Off **Default** Forced Off
Minimum Global Snow Height (float32) Minimum height of the global volumetric snow for the snow trails to be generated.	**Min / Max** 0.000000 - FLOAT_MAX **Default** 0.10000

6.4.8.6 Entities / Lifeforms / Snow Trails / Auto Behavior

Parameter (type)	Values
Surface Contact Distance Limit (float32) Limit for distance between a snow trail source in the **auto** state and the surface below it in meters for them to be considered to be in contact. **NOTE** Use 0.0 to disable checking for the contact with the surface.	**Min / Max** 0.000000 - FLOAT_MAX **Default** 0.5000

6.4.8.7 Entities / Lifeforms / Tracks and Trails

Parameter (type)	Values
Samplers Minimum Detail (uint8) Minimum required sampling detail (in Level-of-Detail units) of samplers utilized by the lifeform tracks subsystem, with 0 being the lowest quality, and 23 being the highest quality.	**Min / Max** 0 - 23 **Default** 0
Samplers Maximum Detail (uint8) Maximum wanted sampling detail (in Level-of-Detail units) of samplers utilized by the lifeform tracks subsystem, with 0 being the lowest quality, and 23 being the highest quality.	**Min / Max** 0 - 23 **Default** 15

6.4.8.8 Entities / Vehicles / Mirrors

Parameter (type)	Values
Near Plane Distance (float32) Near plane distance in meters. If not zero, the near plane distance is used in mirror view calculations. Otherwise the default value inherited from `DefaultViewConfig.xml` or ViewManager Debug UI is used. 0.01 is a recommended value if the default value does not produce desired results, such as visual artifacts in mirrors.	**Min / Max** 0 - FLOAT_MAX **Default** 0.0000

6.4.8.9 Entities / Vehicles / General

Parameter (type)	Values
Show Entity Markings (boolean) Controls the visibility of entity markings (URNs) on vehicles.	**Default** false

6.4.8.10 Entities / Vehicles / Tracer Simulation

Parameter (type)	Values
Affected By Wind (boolean) If true tracers will be affected by wind.	**Default** false
Emits Light (boolean) If true tracers will emit light.	**Default** false
Light Radius (float32) The radius of the light emitted by the tracer. Ignored if EmitsLight is false.	**Default** 10.0000
Light Intensity (float32) The intensity of the light emitted by the tracer. Ignored if EmitsLight is false.	**Default** 3.0000
Light Color R (float32) The red value of the light emitted by the tracer from 0 to 1. Ignored if EmitsLight is false.	**Min / Max** 0 - 1.0000 **Default** 1.0000
Light Color G (float32) The green value of the light emitted by the tracer from 0 to 1. Ignored if EmitsLight is false.	**Min / Max** 0 - 1.0000 **Default** 0.898039
Light Color B (float32) The blue value of the light emitted by the tracer from 0 to 1. Ignored if EmitsLight is false.	**Min / Max** 0 - 1.0000 **Default** 0.007843

6.4.8.11 Entities / Vehicles / Engine

Parameter (type)	Values
Engine Default On **(boolean)** When enabled, vehicles will spawn with their engines turned on.	**Default** true

6.4.8.12 Entities / Vehicles / Tracks

Parameter (type)	Values
Enabled **(boolean)** Whether the tracks drawing system is enabled.	**Default** true
Default Track Enable State **(enum)** Default enable state of tracks drawing for newly registered vehicles. Available options: • 0: Forced Off - Tracks are not drawn. Use if manual control over track drawing is desired. • 1: Auto - Tracks are drawn when conditions defined by the product settings are satisfied.	• Auto • Forced Off **Default** Forced Off
Tracks Update Distance **(float32)** Distance in meters from the last added track point before another track point can be added. • Addition of a track point involves checking the ground material below a vehicle and the vehicle's contact with the ground. • The higher the distance, the better the performance of the tracks drawing subsystem, but the worse the overall tracks quality.	**Min / Max** 0.0000 - FLOAT_MAX **Default** 0.2000
Maximum Updated Vehicles Count **(float32)** Maximum number of vehicles that can have their tracks updated in a single frame. The lower the count, the better the performance of the tracks drawing subsystem, but the worse the overall tracks quality. Use 0 to disable the updated vehicles count limit.	**Min / Max** 0 - UNINT32_MAX **Default** 20

6.4.8.13 Entities / Vehicles / Tracks / Auto Behavior

Parameter (type)	Values
Check Surface Material (boolean) When enabled, material ID of the surface will be checked to see if it allows tracks. When disabled, tracks will be created on all surfaces.	**Default** true
Enable Surface Distance Limit (float64) Maximum distance in meters between the track and the ground for which the track is enabled. If the distance is greater than this value, no track will be generated. Setting this to 0 disables the distance limit.	**Min / Max** 0.0000 - 1000.0000 **Default** 0.50000

6.4.8.14 Entities / Vehicles / Dust Trails

Parameter (type)	Values
Dust Trails Enabled **(boolean)** Whether the dust trails generation system is enabled.	**Default** true
Default Dust Trails Enable State **(enum)** Default enable state of dust trails generation for newly registered vehicles. Available options: • 0: Forced off - The dust trail source is forced to be off. • 1: Auto - The dust trails are generated when conditions defined by the product settings are satisfied.	• Auto • Forced Off **Default** Forced Off
Dust Trails Update Distance **(float32)** Distance in meters from the last dust trail effect update position before another update can be performed. The dust trail effect update involves checking the ground material below a vehicle, the contact of the vehicle with the ground as well as the velocity of the vehicle. The higher the distance, the better the performance of the dust trails generating subsystem, but the worse the overall quality of the dust trail. The update is performed if either of the update distance or the update angle is reached.	**Min / Max** 0.0000 - FLOAT_MAX **Default** 0.2000
Dust Trails Update Angle **(float32)** Angle in degrees from the last dust trail effect update orientation before another update can be performed. The dust trail effect update involves checking the ground material below a vehicle, the contact of the vehicle with the ground as well as the velocity of the vehicle. The higher the angle, the better the performance of the dust trails generating subsystem, but the worse the overall dust trail quality. The update is performed if either the update distance or the update angle is reached.	**Min / Max** 0.0000 - 360.0000 **Default** 9.998114

Parameter (type)	Values
Maximum Updated Vehicles Count (uint32) Maximum number of vehicles that can have their dust trails updated in a single frame. The lower the count, the better the performance of the dust trails generating subsystem, but the worse the overall dust trails quality. Use 0 to disable the updated vehicles count limit.	**Min / Max** 0.0000 - UINT32_MAX **Default** 20.0000
Body Source Dust Trail Effect Scale Multiplier (float32) Multiplier of a dust trail effect scale for body sources. It allows automatic scaling of dust trail effects generated from body sources, which usually replace individual non body sources for better performance. Actual impact of the multiplier depends on the scripted definition of the particle effect.	**Min / Max** 0.0000 - FLOAT_MAX **Default** 1.0000
Minimum Density Required For Generating Dust (float32) Minimum calculated density of a dust trail effect required for its source to generate it.	**Min / Max** 0.0000 - 1.0000 **Default** 0.1000

6.4.8.15 Entities / Vehicles / Dust Trails / Auto Behavior

Parameter (type)	Values
Enabled Sources (enum) Enabled types of dust trail sources when they are in the 'auto' state. Available options: 0: Body - Body dust trail sources. 1: Non-body - Non-body dust trail sources (e.g. wheels and tracks).	• Body • Non-body **Default** Non-body
Surface Contact Distance Limit (float32) Limit for distance between a dust trail source in the **auto** state and the surface below it in meters for them to be considered to be in a contact. Use 0.0 to disable checking for the contact with the surface.	**Min / Max** 0.0000 - FLOAT_MAX **Default** 0.50000

6.4.8.16 Entities / Vehicles / Tracks and Trails

Parameter (type)	Values
Samplers Minimum Detail **(uint8)** Minimum required sampling detail (in Level-of-Detail units) of samplers utilized by vehicle tracks and dust trails subsystems, with 0 being the lowest quality, and 23 being the highest quality.	**Min / Max** 0 - 23 **Default** 0
Samplers Maximum Detail **(uint8)** Maximum wanted sampling detail (in Level-of-Detail units) of samplers utilized by vehicle tracks and dust trails subsystems, with 0 being the lowest quality, and 23 being the highest quality.	**Min / Max** 0 - 23 **Default** 15

6.4.8.17 Entities / Vehicles / Damage

Parameter (type)	Values
Fire Effect **(string)** Particle effect to use when the vehicle is set to be on fire.	**Default** VehicleFireBasic
Smoke Effect **(string)** Particle effect to use when the vehicle is set to be smoking.	**Default** VehicleSmokeBasic
Hit Point Damage Charring **(boolean)** Controls whether vehicle hit points will have a charring effect applied to them to indicate their level of damage.	**Default** true

6.4.8.18 Entities / Vehicles / Smart Scaling

Parameter (type)	Values
Object Types **(enum)** List of object types for which smart scaling should be enabled.	• Land • Air • Surface • Subsurface • Space • Other
Default FOV Horizontal **(float64)** Default Horizontal FOV to do smart scaling calculation against. Views that use a different FOV than the default will be scaled according to how much they differ from the default.	**Min / Max** 0.0000 - FLOAT_MAX **Default** 90.0000
Default FOV Vertical **(float64)** Default Vertical FOV to do smart scaling calculation against. Views that use a different FOV than the default will be scaled according to how much they differ from the default.	**Min / Max** 0.0000 - FLOAT_MAX **Default** 50.6250
Default Pixel Width **(float64)** Pixel width is a scaling constant that will used in smart scaling calculations for all objects. The default pixel width is defined as the pixel width of a 5 meter object viewed from a distance of 4 kilometers. For examples, a default pixel width of '4.75' will make a 5 meter object that is 4 kilometers from the view appear on screen as approximately 4.75 pixels wide.	**Min / Max** 0.0000 - FLOAT_MAX **Default** 4.0000
Camera Move Threshold **(float64)** How far a view must move for objects in its frustum to recalculate smart scaling sizes.	**Min / Max** 0.0000 - FLOAT_MAX **Default** 1.0000
Use Viewport Scaling **(boolean)** Flag indicating if view ports should be taken into account when using smart scaling.	**Default** false

6.4.8.19 Entities / Vehicles / Animation Culling

Parameter (type)	Values
Max Force Update Distance (float64) If the distance from the camera to a vehicle is less than this value, an animation update will be forced for that vehicle.	**Min / Max** 0.0000 - 1000.0000 **Default** 10.0000
Min Skip Update Distance (float64) If the distance from the camera to a vehicle is greater than this value, the animation update will be skipped for that vehicle.	**Min / Max** 0.0000 - 1000.0000 **Default** 260.0000

6.4.8.20 Entities / Vehicles / Lights

Parameter (type)	Values
Spotlight Intensity Multiplier (float32) Multiplier for the intensity of vehicle spotlights (typically head lights on ground vehicles).	**Min / Max** 0.0000 - 10000.0000 **Default** 1.0000
Pointlight Intensity Multiplier (float32) Multiplier for the intensity of vehicle point lights (typically marker lights on aircraft).	**Min / Max** 0.0000 - 10000.0000 **Default** 1.0000
Spotlight Flare Scale Multiplier (float32) Multiplier for the flare scale of vehicle spotlights (typically headlights on ground vehicles).	**Min / Max** 0.0000 - 10000.0000 **Default** 1.0000
Pointlight Flare Scale Multiplier (float32) Multiplier for the flare scale of vehicle point lights (typically marker lights on aircraft).	**Min / Max** 0.0000 - 10000.0000 **Default** 1

6.4.8.21 Entities / Vehicles / Animations / Default Animations / Animation

Controls the default relative phase value of an animation source when a model with that animation source is created.

> **NOTE**
> Changing this setting during runtime will only affect newly created models.

Parameter (type)	Values
Source Name **(string)** The name of the animation source that the value is to be set for.	**Default** (EmptyString)
Relative Phase Value **(float32)** The relative phase value to set for the animation source. A relative phase describes a phase value relative to the animation source's minimum and maximum values, where 0 is the minimum phase and 1 is the maximum phase.	0.00000 - 10000.00 **Default:** 0.000000

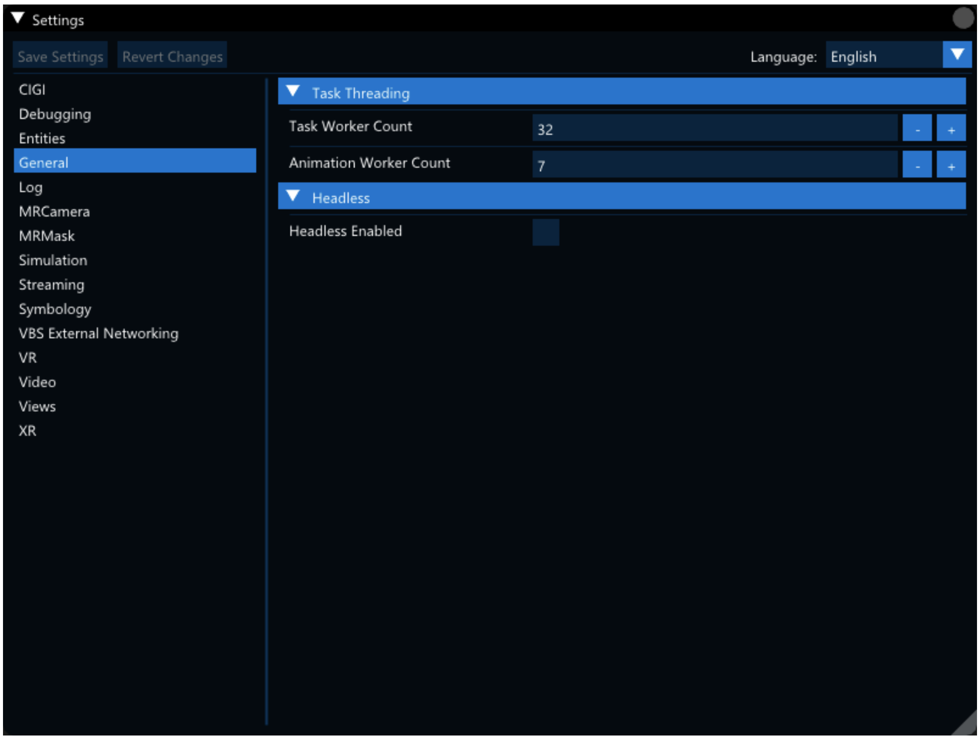
6.4.8.22 Entities / Vehicles / Animations / Automatic Animations

Parameter (type)	Values
Wheel and Track Rotation (boolean) Controls whether vehicles have automatic wheel and track rotation animations enabled by default. The automatic rotation is simulated based on the vehicle's forward velocity. Disable if such animations are not required, or manual control over animations is desired.	**Default** true
Steering (boolean) Controls whether vehicles have automatic wheel steering animations enabled by default. The automatic steering is simulated based on the vehicles yaw angular velocity. Disable if such animations are not required, or manual control over animations is desired.	**Default** true
Suspension (boolean) Controls whether vehicles have automatic suspension animations enabled by default. The automatic suspension is simulated based on the vehicles height above the ground. Disable if such animations are not required, or manual control over animations is desired.	**Default** true
Time (boolean) Controls whether time-based animations are automatically simulated by default. When enabled, entity animations controlled by the simulation and clock time will be automatically updated. Disable if such animations are not required, or manual control over animations is desired.	**Default** true
Wind Global (boolean) Controls whether wind velocity-based animations are automatically simulated by default. When enabled, entity animations controlled by the wind will be automatically updated.	**Default** true

6.4.9 General Settings

Customize General settings for a simulation host using either of the following methods:

- **Settings UI** - Press **Tab** to open the VBS Blue IG Settings (on page 201) and make adjustments, as required.



- **Edit XML file** - Open the `General.xml` file, located in the following directory and edit, as required:

  `%LOCALAPPDATA%\Bohemia Interactive Simulations\VBS Blue IG\`*`version`*`\Settings\`

The tables below describe the type and purpose for each setting.

6.4.9.1 General / Current Locale

Parameter (type)	Values
Language **(string)** The Locale Code for the Language that will be used. Locale Codes can be up to 3 letters.	**Default** (EmptyString)

6.4.9.2 General / Task Threading

Parameter (type)	Values
Task Worker Count **(uint32)** Sets the number of task workers that get created. Task workers run tasks concurrently. Minimum 2, but should be increased for every task that does not join or finish.	**Min / Max** 2 - UINT32_MAX **Default** 32
Animation Worker Count **(uint32)** Number of threads that should be used for character animation processing.	**Min / Max** 1 - 64 **Default** 7

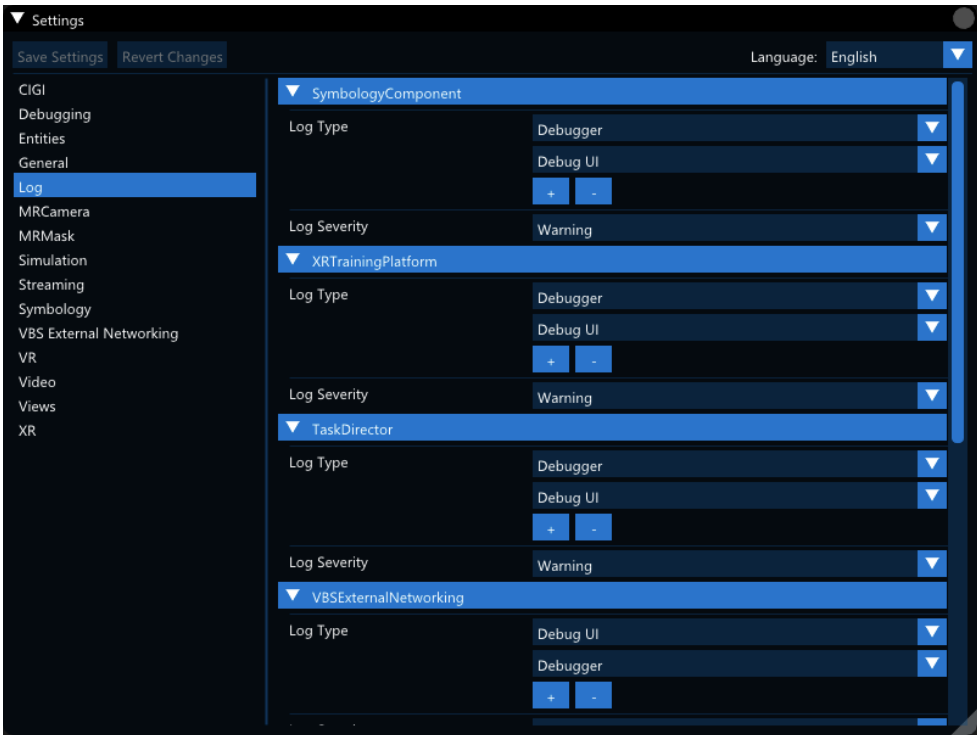
6.4.9.3 General / Headless

Parameter (type)	Values
Headless Enabled **(boolean)** If True, the client runs in headless mode. Headless mode will run VBS Blue IG without rendering.	**Default** false

6.4.10 Log Settings

Customize Log settings for a simulation host using either of the following methods:

- **Settings UI** - Press **Tab** to open the VBS Blue IG Settings (on page 201) and make adjustments, as required.



- **Edit XML file** - Open the `Log.xml` file, located in the following directory and edit, as required:

  `%LOCALAPPDATA%\Bohemia Interactive Simulations\VBS Blue IG\` `*version*\Settings\`

The tables below describe the type and purpose for each setting.

6.4.10.1 Log / SymbologyComponent

Parameter (type)	Values
Log Type **(enum)** Supported log outputs. May support multiple values.	• Off • File • Console • Debugger • DebugUI **Default** • Debugger • DebugUI
Log Severity **(enum)** Minimum logging severity level.	• Critical • Error • Warning • Information • Debug **Default** Warning

6.4.10.2 Log / XRTrainingPlatform

Parameter (type)	Values
Log Type **(enum)** Supported log outputs. May support multiple values.	• Off • File • Console • Debugger • DebugUI **Default** • Debugger • DebugUI
Log Severity **(enum)** Minimum logging severity level.	• Critical • Error • Warning • Information • Debug **Default** Warning

6.4.10.3 Log / TaskDirector

Parameter (type)	Values
Log Type **(enum)** Supported log outputs. May support multiple values.	• Off • File • Console • Debugger • DebugUI **Default** • Debugger • DebugUI
Log Severity **(enum)** Minimum logging severity level.	• Critical • Error • Warning • Information • Debug **Default** Warning

6.4.10.4 Log / VBSExternalNetworking

Parameter (type)	Values
Log Type **(enum)** Supported log outputs. May support multiple values.	• Off • File • Console • Debugger • DebugUI **Default** • Debugger • DebugUI
Log Severity **(enum)** Minimum logging severity level.	• Critical • Error • Warning • Information • Debug **Default** Warning

6.4.10.5 Log / CigiProtocol

Parameter (type)	Values
Log Type **(enum)** Supported log outputs. May support multiple values.	• Off • File • Console • Debugger • DebugUI **Default** • Debugger • DebugUI
Log Severity **(enum)** Minimum logging severity level.	• Critical • Error • Warning • Information • Debug **Default** Warning

6.4.10.6 Log / BlueIGBlueSystems

Parameter (type)	Values
Log Type **(enum)** Supported log outputs. May support multiple values.	• Off • File • Console • Debugger • DebugUI **Default** • Debugger • DebugUI
Log Severity **(enum)** Minimum logging severity level.	• Critical • Error • Warning • Information • Debug **Default** Warning

6.4.10.7 Log / Blue

Parameter (type)	Values
Log Type **(enum)** Supported log outputs. May support multiple values.	• Off • File • Console • Debugger • DebugUI **Default** • Debugger • DebugUI
Log Severity **(enum)** Minimum logging severity level.	• Critical • Error • Warning • Information • Debug **Default** Critical

6.4.10.8 Log / IGMultiChannel

Parameter (type)	Values
Log Type **(enum)** Supported log outputs. May support multiple values.	• Off • File • Console • Debugger • DebugUI **Default** • Debugger • DebugUI
Log Severity **(enum)** Minimum logging severity level.	• Critical • Error • Warning • Information • Debug **Default** Warning
Verbose Logging **(boolean)** Enables / disables verbose logging.	**Default** false

6.4.10.9 Log / ViewManager

Parameter (type)	Values
Log Type **(enum)** Supported log outputs. May support multiple values.	• Off • File • Console • Debugger • DebugUI **Default** • Debugger • DebugUI
Log Severity **(enum)** Minimum logging severity level.	• Critical • Error • Warning • Information • Debug **Default** Warning
Verbose Logging **(boolean)** Enables / disables verbose logging.	**Default** false

6.4.10.10 Log / WarpComponent

Parameter (type)	Values
Log Type **(enum)** Supported log outputs. May support multiple values.	• Off • File • Console • Debugger • DebugUI **Default** • Debugger • DebugUI
Log Severity **(enum)** Minimum logging severity level.	• Critical • Error • Warning • Information • Debug **Default** Warning

6.4.10.11 Log / CrashReporter

Parameter (type)	Values
Log Type **(enum)** Supported log outputs. May support multiple values.	• Off • File • Console • Debugger • DebugUI **Default** • Debugger • DebugUI
Log Severity **(enum)** Minimum logging severity level.	• Critical • Error • Warning • Information • Debug **Default** Warning

6.4.10.12 Log / LegacyPlugin

Parameter (type)	Values
Log Type **(enum)** Supported log outputs. May support multiple values.	• Off • File • Console • Debugger • DebugUI **Default** • Debugger • DebugUI
Log Severity **(enum)** Minimum logging severity level.	• Critical • Error • Warning • Information • Debug **Default** Warning

6.4.10.13 Log / VideoStreaming

Parameter (type)	Values
Log Type **(enum)** Supported log outputs. May support multiple values.	• Off • File • Console • Debugger • DebugUI **Default** Debugger
Log Severity **(enum)** Minimum logging severity level.	• Critical • Error • Warning • Information • Debug **Default** Warning

6.4.10.14 Log / BlueIGEntitySystems

Parameter (type)	Values
Log Type **(enum)** Supported log outputs. May support multiple values.	• Off • File • Console • Debugger • DebugUI **Default** • Debugger • DebugUI
Log Severity **(enum)** Minimum logging severity level.	• Critical • Error • Warning • Information • Debug **Default** Warning

6.4.10.15 Log / BlueIGWorldSystems

Parameter (type)	Values
Log Type **(enum)** Supported log outputs. May support multiple values.	• Off • File • Console • Debugger • DebugUI **Default** • Debugger • DebugUI
Log Severity **(enum)** Minimum logging severity level.	• Critical • Error • Warning • Information • Debug **Default** Warning

6.4.10.16 Log / BlueIGDiagnosticSystems

Parameter (type)	Values
Log Type **(enum)** Supported log outputs. May support multiple values.	• Off • File • Console • Debugger • DebugUI **Default** • Debugger • DebugUI
Log Severity **(enum)** Minimum logging severity level.	• Critical • Error • Warning • Information • Debug **Default** Warning

6.4.10.17 Log / EntityDirector

Parameter (type)	Values
Log Type **(enum)** Supported log outputs. May support multiple values.	• Off • File • Console • Debugger • DebugUI **Default** • Debugger • DebugUI
Log Severity **(enum)** Minimum logging severity level.	• Critical • Error • Warning • Information • Debug **Default** Warning

6.4.10.18 Log / RopeSystem

Parameter (type)	Values
Log Type **(enum)** Supported log outputs. May support multiple values.	• Off • File • Console • Debugger • DebugUI **Default** • Debugger • DebugUI
Log Severity **(enum)** Minimum logging severity level.	• Critical • Error • Warning • Information • Debug **Default** Warning

6.4.10.19 Log / BlueVR

Parameter (type)	Values
Log Type **(enum)** Supported log outputs. May support multiple values.	• Off • File • Console • Debugger • DebugUI **Default** • Debugger • DebugUI
Log Severity **(enum)** Minimum logging severity level.	• Critical • Error • Warning • Information • Debug **Default** Warning

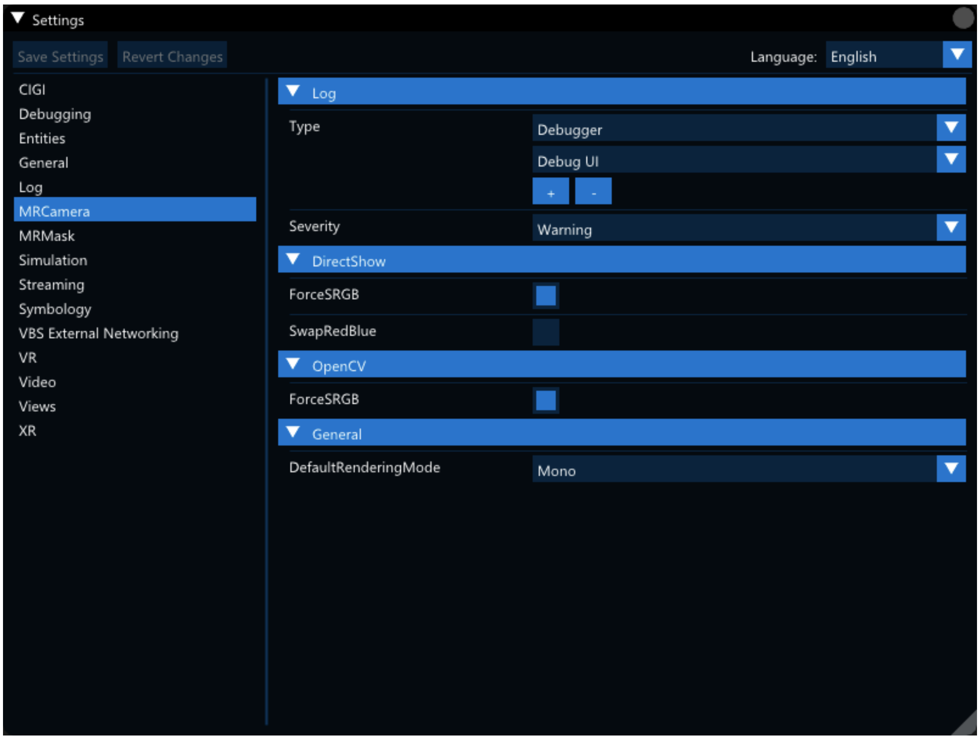
6.4.10.20 Log / DebugUI

Parameter (type)	Values
Log Type **(enum)** Supported log outputs. May support multiple values.	• Off • File • Console • Debugger • DebugUI **Default** • Debugger • DebugUI
Log Severity **(enum)** Minimum logging severity level.	• Critical • Error • Warning • Information • Debug **Default** Warning

6.4.11 MRCamera Settings

Customize Mixed Reality (MR) Camera settings for a simulation host using either of the following methods:

- **Settings UI** - Press **Tab** to open the VBS Blue IG Settings (on page 201) and make adjustments, as required.



- **Edit XML file** - Open the `MRCamera.xml` file, located in the following folder and edit, as required:

  `%LOCALAPPDATA%\Bohemia Interactive Simulations\VBS Blue IG\`*`version`*`\Settings\`

The following tables describe the type and purpose for each setting:

6.4.11.1 MRCamera / Log

Logging options specific to Mixed-Reality Camera.

Parameter (type)	Values
Type **(enum)** Supported log outputs. May support multiple values.	• Off • File • Console • Debugger • DebugUI **Default** • Debugger • DebugUI
Severity **(enum)** Minimum logging severity level.	• Critical • Error • Warning • Information • Debug **Default** Warning

6.4.11.2 MRCamera / DirectShow

Options for DirectShow imagery.

Parameter (type)	Values
ForceSRGB **(boolean)** Forces sRGB conversion for DirectShow imagery.	**Default** true
SwapRedBlue **(boolean)** Swaps red and blue color components (RGB->BGR) for DirectShow imagery.	**Default** false

6.4.11.3 MRCamera / OpenCV

Options for OpenCV imagery.

Parameter (type)	Values
ForceSRGB **(boolean)** Forces sRGB conversion for OpenCV imagery.	**Default** true

6.4.11.4 MRCamera / General

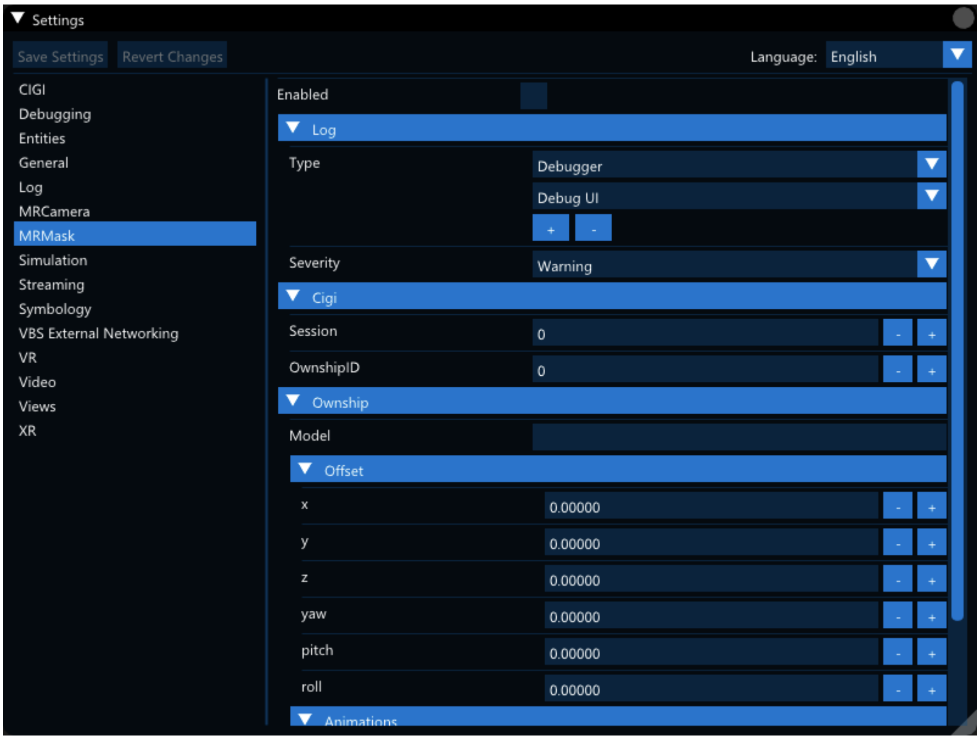
General options.

Parameter (type)	Values
DefaultRenderingMode **(enum)** Default rendering mode. Mono (default), or Stereo side-by-side.	• Mono • Stereo_SBS **Default** Mono

6.4.12 MRMask Settings

Customize Mixed Reality (MR) Mask settings for a simulation host using either of the following methods:

- **Settings UI** - Press **Tab** to open the VBS Blue IG Settings (on page 201) and make adjustments, as required.



- **Edit XML file** - Open the `MRMask.xml` file, located in the following folder and edit, as required:

  `%LOCALAPPDATA%\Bohemia Interactive Simulations\`
  `VBS Blue IG\`*`version`*`\Settings\`

For more information about the component, see MRMasking Component (on page 76).

The following tables describe the type and purpose for each setting:

6.4.12.1 MRMask

Mixed-Reality Mask settings for VBS Blue IG.

Parameter (type)	Values
Enabled (boolean) Toggles the MR Mask component.	**Default** false

6.4.12.2 MRMask / Log

Logging options specific to MR Mask.

Parameter (type)	Values
Type **(enum)** Supported log outputs. May support multiple values.	• Off • File • Console • Debugger • DebugUI **Default** • Debugger • DebugUI
Severity **(enum)** Minimum logging severity level.	• Critical • Error • Warning • Information • Debug **Default** Warning

6.4.12.3 MRMask / Cigi

MR Mask CIGI connectivity settings.

Parameter (type)	Values
Session **(uint8)** Session ID for host system, in case multiple sessions are configured in CigiProtocol.	**Default** 0
OwnshipID **(uint8)** CIGI ID for ownship entity. This is typically 0 for most CIGI hosts.	**Default** 0

6.4.12.4 MRMask / Ownship

Ownship settings.

Parameter (type)	Values
Model **(string)** Model name (shape path or classname) for the ownship class if host doesn't create a model. This should be blank for hosts that create a visual entity for the ownship.	**Default** (EmptyString)

6.4.12.5 MRMask / Ownship / Offset

MR Mask ownship translation offset, in case the host uses different model origin than ours (e.g. center of gravity, etc).

Parameter (type)	Values
x **(float64)**	**Default** 0
y **(float64)**	**Default** 0
z **(float64)**	**Default** 0
yaw **(float64)**	**Min / Max** -180.000 - 180.000 **Default** 0
pitch **(float64)**	**Min / Max** -180.000 - 180.000 **Default** 0
roll **(float64)**	**Min / Max** -180.000 - 180.000 **Default** 0

6.4.12.6 MRMask / Ownship / Animations

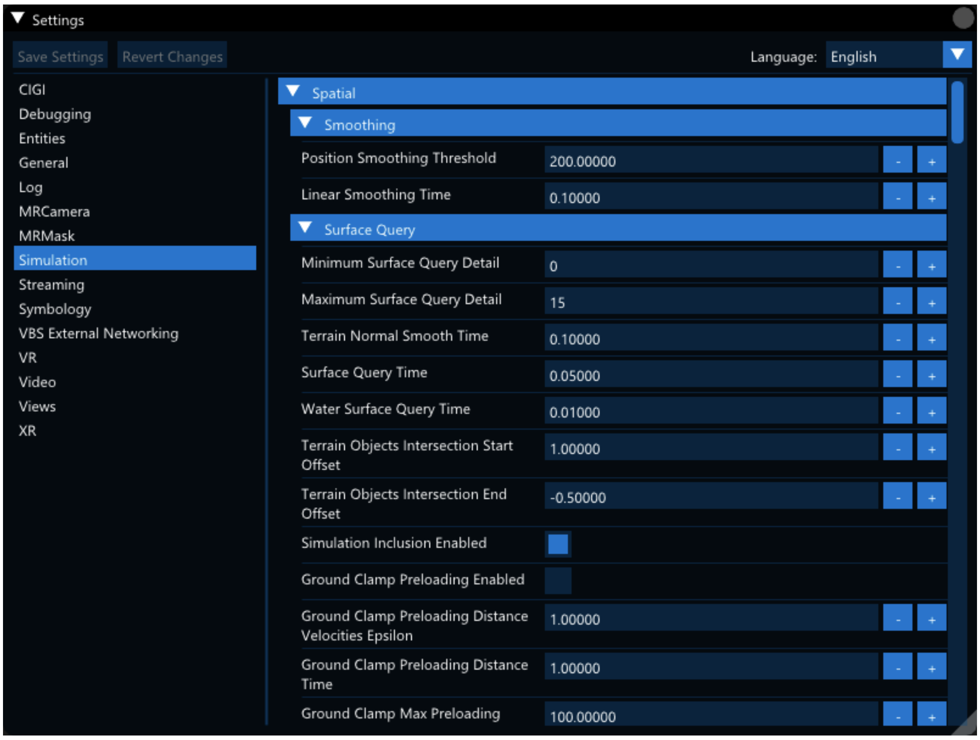
Special ownship animation logic.

Parameter (type)	Values
Rotorcraft **(boolean)** Is Ownship a Rotor Aircraft	**Default** false
RotorRPM **(float32)** Rotor RPM	**Default** 278.000
TailRotorRPM **(float32)** Tail Rotor RPM	**Default** 1500.000

6.4.13 Simulation Settings

Customize Simulation settings for a simulation host using either of the following methods:

- **Settings UI** - Press **Tab** to open the VBS Blue IG Settings (on page 201) and make adjustments, as required.



- **Edit XML file** - Open the `Simulation.xml` file, located in the following directory and edit, as required:

  `%LOCALAPPDATA%\Bohemia Interactive Simulations\VBS Blue IG\`*`version`*`\Settings\`

The tables below describe the type and purpose for each setting.

6.4.13.1 Simulation / Spatial

Parameter (type)	Values
Num Dispatch Tasks (uint32) Number of task threads to dispatch when performing parallel spatial processing.	**Min / Max** 1 - 32 **Default** 4

6.4.13.2 Simulation / Spatial / Smoothing

Parameter (type)	Values
Position Smoothing Threshold **(float32)** Distance in meters a new position update has to be below from the original position for smoothing to occur.	**Min / Max** 0.0000 - FLOAT_MAX **Default** 200.0000
Linear Smoothing Time **(float32)** The amount of time in seconds a smoothing process will take to smooth towards the target position/orientation when using linear smoothing. Specifying 0 disables smoothing.	**Min / Max** 0.0000 - FLOAT_MAX **Default** 0.10000

6.4.13.3 Simulation / Spatial / Surface Query

Parameter (type)	Values
Minimum Surface Query Detail (uint8) Specifies the minimum required detail for surface queries. Value represents LOD, with 0 being lowest quality, and 23 being the highest quality.	**Min / Max** 0 - 23 **Default** 0
Maximum Surface Query Detail (uint8) Specifies the maximum required detail for surface queries. Value represents LOD, with 0 being lowest quality, and 23 being the highest quality.	**Min / Max** 0 - 23 **Default** 15
Terrain Normal Smooth Time (float32) The amount of time in seconds for linear smoothing of terrain normals. Set to 0 to disable smoothing.	**Min / Max** 0.0000 -FLOAT_MAX **Default** 0.10000
Surface Query Time (float32) Specifies the amount of time in seconds that must pass before a surface query is repeated due to insufficient LOD detail. Note that too high a value may cause the entity to update slowly when moving across surfaces.	**Min / Max** 0.0000 -FLOAT_MAX **Default** 0.05000
Water Surface Query Time (float32) Specifies the amount of time in seconds that must pass before repeating a query in the water when water clamping is enabled. Due to constant changes in the water surface, the query will continue to repeat as long as an entity is in the water. Note that too high a value may cause the entity to update slowly when above water surfaces.	**Min / Max** 0.0000 -FLOAT_MAX **Default** 0.01000
Terrain Objects Intersection Start Offset (float64) Start offset for finding terrain objects during ground clamping. The offset is always positive and represents a distance above the clamped entity, where the raycasting starts.	**Min / Max** 0.0000 -DOUBLE_MAX **Default** 1.0000
Terrain Objects Intersection End Offset (float64) End offset for finding terrain objects during ground clamping. The offset is always negative and represents a distance below the clamped entity, where the raycasting ends.	**Min / Max** -DOUBLE_MAX0 - 0.0000 **Default** -0.5000

Parameter (type)	Values
Simulation Inclusion Enabled (boolean) Specifies whether terrain surface simulation (e.g. water waves) should be included for new entities until explicitly overridden.	**Default** true
Ground Clamp Preloading Enabled (boolean) Specifies whether data preloading is enabled for ground clamping purposes. This will cause the ground clamping process to load additional terrain data in a radius around the ground clamped entity, based on the entity's velocity. This will improve ground clamping accuracy, at the cost of additional load on the system.	**Default** false
Ground Clamp Preloading Distance Velocities Epsilon (float64) Specifies minimum change of an entity's velocity in order for the preloading distance for ground clamping purposes for the entity is recalculated and reconfigured.	**Min / Max** 0.0000 -DOUBLE_MAX **Default** 1.0000
Ground Clamp Preloading Distance Time (float64) Determines distance from entities to preload data around moving entities for ground clamping purposes, based on their velocities. For example if an entity is moving 2 m/s and the setting value is 10 seconds, the data will be preloaded in 20 meters radius from the entity, which is the distance the entity would travel over the configured time assuming its velocity remained unchanged.	**Min / Max** 0.0000 -DOUBLE_MAX **Default** 1.0000
Ground Clamp Max Preloading Distance (float64) Maximum distance from the entity the data are preloaded for the ground clamping purposes..	**Min / Max** 0.01000 -DOUBLE_MAX **Default** 100
Ground Clamp Min Preloading Distance (float64) Minimum distance from the entity the data are preloaded for the ground clamping purposes.	**Min / Max** 0.01000 -DOUBLE_MAX **Default** 0.01000

6.4.13.4 Simulation / Queries / Samplers

Parameter (type)	Values
Force Sampler Settings (boolean) If enabled, forces all samplers to use the settings from here instead of their own configuration.	**Default** false
Minimum Detail (uint8) Required minimum of the detail range to sample data in. It is specified as a Level-Of-Detail (LOD) value, with 0 being the lowest quality, and 23 being the highest quality.	**Min / Max** 0 -23 **Default** 0
Maximum Detail (uint8) Wanted maximum of the detail range to sample data in. It is specified as a Level-Of-Detail (LOD) value, with 0 being the lowest quality, and 23 being the highest quality.	**Min / Max** 0 -23 **Default** 23
Search Strategy (enum) Strategy to follow while sampling data.	• Bottom-up Search Strategy • Progressive Search Strategy • Automatic Search Strategy **Default** Automatic Search Strategy
Preloading Enabled (boolean) If enabled, sampler will asynchronously preload area with a specific radius up to a specific detail. Those can be set with "Preloading Radius" and "Preloading Detail" settings.	**Default** false
Preloading Radius (float64) Radius of the area to preload if preloading is enabled.	**Min / Max** 0.0000 - DOUBLE_MAX **Default** 1.0000

Parameter (type)	Values
Preloading Detail **(uint8)** Detail to preload the area up to if preloading is enabled. It is specified as a Level-Of-Detail (LOD) value, with 0 being the lowest quality, and 23 being the highest quality.	**Min / Max** 0 -23 **Default** 23
Include Simulation **(boolean)** If enabled, simulation effects are considered while sampling. Examples of the affected features are water surface position and normal, which may change over time as a result of waves simulation.	**Default** false
Object Sampling Min Height **(float64)** Lower height limit of object data sampling range. Only objects within the range are considered while sampling. It is specified in meters and represents height above (if positive) or below (if negative) the globe ellipsoid surface.	**Default** -11000.0000
Object Sampling Max Height **(float64)** Upper height limit of object data sampling range. Only objects within the range are considered while sampling. It is specified in meters and represents height above (if positive) or below (if negative) the globe ellipsoid surface.	**Default** 9000.0000
Object Sampling Max Intersections Count **(uint32)** Maximum number of intersections checked during the object data sampling to find an acceptable one, before declaring the object data sample query operation unsuccessful. The value must be greater than 0.	1 - UINT32_MAX **Default** 64

6.4.13.5 Simulation / Queries / Terrain

Parameter (type)	Values
Material Test Offset **(float64)** The offset multiplier used for terrain intersections when querying for the terrain surface material. A higher number will result in a larger intersection line being calculated, which affects performance but may increase accuracy when calculating on steep terrain.	**Min / Max** 0 - 0 **Default** 0.20000

6.4.13.6 Simulation / Queries / Intersection Type Override

Enable or disable specific intersection query types, overriding what other components are requesting. If no types are enabled, all intersection is disabled.

Parameter (type)	Values
Streamed Object (boolean)	**Default** true
Dynamic Object (boolean)	**Default** true
Geometry (boolean)	**Default** true
Biome Tree (boolean)	**Default** true
Biome Bush (boolean)	**Default** true
Ground (boolean)	**Default** true
Water (boolean)	**Default** true
Particles (boolean)	**Default** true
Volume (boolean)	**Default** true

6.4.13.7 Simulation / Queries / Entity Surface

Contains settings related to EntitySurface queries.

Parameter (type)	Values
Object Intersection Enabled (boolean) Determines whether objects can be intersected when performing EntitySurface queries. Setting this to False will disable object intersections.	**Default** true

6.4.13.8 Simulation / Thermal / General

Parameter (type)	Values
Default Vehicle Tracks Temperature (enum) Default temperature of vehicle wheels / tracks when spawned.	• Cold • Hot **Default** Cold

6.4.13.9 Simulation / Thermal / Simulation

Parameter (type)	Values
Enabled (boolean) When enabled, VBS Blue IG will simulate thermal values if they are not set by a host.	**Default** false

6.4.13.10 Simulation / Thermal / Simulation / Damage

Parameter (type)	Values
Burning Engine Factor (float32) Temperature multiplier of a vehicle which is on fire (6 by default).	**Min / Max** 0.00100 - 60 **Default** 6.0000
Burning Engine Minutes to Heat Up (float32) The time it takes to reach max heat while a vehicle is on fire (in minutes).	**Min / Max** 0.00100 - 60.0000 **Default** 5.0000

6.4.13.11 Simulation / Thermal / Simulation / Speed

Parameter (type)	Values
Min Speed to Heat Up **(float32)** Minimum speed a unit must be moving to heat up (in m/s).	**Min / Max** 0.00100 - 60.0000 **Default** 1.0000

6.4.13.12 Simulation / Thermal / Simulation / Engine

Parameter (type)	Values
Engine Minutes to Heat Up **(float32)** The time it takes for the engine to heat up when turned on (in minutes).	**Min / Max** 0.00100 - 60.0000 **Default** 10.0000
Engine Minutes to Cool Down **(float32)** The time it takes for the engine to cool down when turned off (in minutes).	**Min / Max** 0.00100 - 60.0000 **Default** 30.0000

6.4.13.13 Simulation / Thermal / Simulation / Wheel Movement

Parameter (type)	Values
Wheels Minutes to Heat Up **(float32)** The time it takes for wheels / entities to heat up when moving (in minutes).	**Min / Max** 0.00100 - 60.0000 **Default** 2.0000
Wheels Minutes to Cool Down **(float32)** The time it takes for wheels/ entities to cool down when not moving (in minutes).	**Min / Max** 0.00100 - 60.0000 **Default** 10.0000

6.4.13.14 Simulation / Thermal / Simulation / Barrel

Parameter (type)	Values
Barrel Heat Factor **(float32)** The amount a turret / gun heats up from a single shot as a percent (values between 0 and 1).	**Min / Max** 0.00100 - 1.0000 **Default** 0.03300
Barrel Minutes to Cool Down **(float32)** The time it takes for turrets/guns to cool down after firing (in minutes).	**Min / Max** 0.001 - 60 **Default** 5.0000

6.4.13.15 Simulation / Thermal / Simulation / Lifeforms

Parameter (type)	Values
Lifeform Minutes to Cool Down **(float32)** The time it takes for lifeforms to cool down when dead (in minutes).	**Min / Max** 0.00100 - 60.0000 **Default** 10.0000
Tracers Always Hot **(boolean)** Whether tracer models should always display as hot in thermal imaging sensors. When enabled, tracers will always be 100% hot. When disabled, tracer temperature must be controlled by a host.	**Default** true
Hot Object Paths **(string)** List of model paths of simple objects that should always show as hot in thermal imaging sensors.	**Default** • vbs2\weapons\data\bullettracer\tracer_red.p3d • vbs2\weapons\data\bullettracer\tracer_green.p3d • vbs2\weapons\data\bullettracer\tracer_yellow.p3d • vbs2\weapons\data\bullettracer\tracer_white.p3d

6.4.13.16 Simulation / Environment / Default Weather / Time

Parameter (type)	Values
Universal Time Increment Modifier (float64) The modifier for universe time incrementing. Default is 1 for a normal rate of incrementing. Set to 0 to freeze the time of day. Set to a number > 1 to speed up time of day progression, or a number (0-1) to slow down time of day progression. Set to a negative number to reverse time of day progression.	0.000000 - DOUBLE_MAX **Default** 1.0000
Simulation Time Increment Modifier (float64) The modifier for environmental simulation time incrementing. Default is 1 for a normal rate of incrementing. Set to 0 to freeze the environmental simulation. Set to a number > 1 to speed up environmental simulation progression, or a number (0-1) to slow down environmental simulation progression. Set to a negative number to reverse environmental simulation progression.	0.000000 - DOUBLE_MAX **Default** 1.0000
Effect Time Increment Modifier (float64) The modifier for environmental effect time incrementing. Default is 1 for a normal rate of incrementing. Set to 0 to freeze the environmental effects. Set to a number > 1 to speed up effect progression, or a number (0-1) to slow down effect progression. Set to a negative number to reverse effect progression.	0.000000 - DOUBLE_MAX **Default** 1.0000
Adaptation Time Increment Modifier (float64) The modifier for environmental adaptation time incrementing. Default is 1 for a normal rate of incrementing. Set to 0 to freeze the environmental adaptations. Set to a number > 1 to speed up adaptation progression, or a number (0-1) to slow down adaptation progression. Set to a negative number to reverse adaptation progression.	0.000000 - DOUBLE_MAX **Default** 1.0000

6.4.13.17 Simulation / Environment / Default Weather / Global Precipitation

Parameter (type)	Values
Type **(enum)** The type of visual precipitation effect. There can only be one type of effect active at any one time.	**Value** • Rain • Snow • Sleet • Hail **Default** Rain
Density **(float32)** The precipitation density [0-1]. Set to 0 to disable precipitation.	**Min / Max** 0.0000 - 1.0000 **Default** 0.0000
Severity **(float32)** The precipitation severity (speed of precipitation) [0-1]. Set to 0 to disable precipitation.	**Min / Max** 0.0000 - 1.0000 **Default** 0.0000
Max Altitude **(float32)** The altitude in meters above the ellipsoid at which precipitation stops.	**Default** 20000.0000
Puddle Size **(float32)** Current size of puddles [0-1]. 0 being no puddles.	**Min / Max** 0 - 1 **Default** 0.0000
Wetness **(float32)** Intensity of darkening and reflectivity effect which causes surfaces to appear wet from rain.	**Min / Max** 0.0000 - 1.0000 **Default** 0.0000
Snow Amount (Ground) **(float32)** Snow coverage on ground relative to default level in meters. Negative values will remove snow from areas which have snow coverage by default.	**Min / Max** -1.0000 - 20.0000 **Default** 0.0000

Parameter (type)	Values
Snow Density **(float32)** Snow density g/cm^3. Valid range [0.0,1]. Density of fresh fallen snow is around 0.1. Solid ice can have density as high as 0.9.	**Min / Max** 0.010000 - 1.000000 **Default** 0.2000
Snow Amount (Trees) **(float32)** Snow coverage on trees [0-1]. 0 being no snow.	**Min / Max** 0.0000 - 1.0000 **Default** 0.0000
Snow Amount (Buildings) **(float32)** Snow coverage on buildings [0-1]. 0 being no snow.	**Min / Max** 0.0000 - 1.0000 **Default** 0.0000
Snow Line Altitude **(float32)** Altitude in meters, relative to the WGS84 ellipsoid, where snow begins to appear. This setting is ignored if **Snow Transition Band Thickness** is negative.	**Default** 0.0000
Snow Transition Band Thickness **(float32)** Altitude offset to **Snow Line Altitude**, where snow reaches full coverage. Creates a blend region between [Snow Line Altitude, Snow Line Altitude + Snow Transition Band Thickness], in which snow gradually appears. Set to a negative value to show snow at all altitudes.	**Default** -1.0000
Freeze Water Surfaces **(boolean)** Determines whether all water surfaces are in a liquid state.	**Default** false

6.4.13.18 Simulation / Environment / Default Weather / Global Snow Plowing

Parameter (type)	Values
Depth **(float32)** The depth in meters for snow plowing on roads. For example, if there is 1m of snow: - 0.75m of snow plowing depth will result in the road being covered in 0.25m of snow.- 1m of snow plowing depth will result in roads being barely covered by snow. - 2m of snow plowing depth will clear more snow from the road and widen the cleared area.	**Min / Max** 0.0000 - 20.0000 **Default** 0.0000

6.4.13.19 Simulation / Environment / Default Weather / Global Haze

Parameter (type)	Values
Type **(enum)** The type of the default global haze effect.	**Values** • Fog • Sand • Dust • Custom **Default** Fog
Visibility **(float32)** Visibility range through the haze in meters. Set to 0 to disable haze.	**Min / Max** 0.0000 - 2000.0000 **Default** 95000.0000
Base Altitude **(float32)** The altitude in meters above the ellipsoid at which the given visibility is used.	**Min / Max** -FLOAT_MAX - FLOAT_MAX **Default** 0.0000
Altitude Scale **(float32)** How fast the visibility changes with altitude. The value given means how much you need to ascend (in meters) for the visibility of the fog to double.	**Min / Max** -FLOAT_MAX - FLOAT_MAX **Default** 500.0000

6.4.13.20 Simulation / Environment / Default Weather / Global Haze / Haze Color Fog

Parameter (type)	Values
Red **(float32)** The red component of the color.	**Min / Max** 0.0000 - 1.0000 **Default** 1.0000
Green **(float32)** The green component of the color.	**Min / Max** 0.0000 - 1.0000 **Default** 1.0000
Blue **(float32)** The blue component of the color.	**Min / Max** 0.0000 - 1.0000 **Default** 1.0000

6.4.13.21 Simulation / Environment / Default Weather / Global Haze / Haze Color Sand

Parameter (type)	Values
Red **(float32)** The red component of the color.	**Min / Max** 0.0000 - 1.0000 **Default** 0.96000
Green **(float32)** The green component of the color.	**Min / Max** 0.0000 - 1.0000 **Default** 0.89000
Blue **(float32)** The blue component of the color.	**Min / Max** 0.0000 - 1.0000 **Default** 0.8000

6.4.13.22 Simulation / Environment / Default Weather / Global Haze / Haze Color Dust

Parameter (type)	Values
Red **(float32)** The red component of the color.	**Min / Max** 0.0000 - 1.0000 **Default** 0.96000
Green **(float32)** The green component of the color.	**Min / Max** 0.0000 - 1.0000 **Default** 0.94000
Blue **(float32)** The blue component of the color.	**Min / Max** 0.0000 - 1.0000 **Default** 0.91000

6.4.13.23 Simulation / Environment / Default Weather / Local Haze

Parameter (type)	Values
Transition Time **(float64)** The time in seconds it takes to smoothly transition between global and local haze. Set to 0 for an instant transition.	**Min / Max** 0.0000 - 60.0000 **Default** 0.75000

6.4.13.24 Simulation / Environment / Default Weather / Global Water

Parameter (type)	Values
Sea State **(float32)** The sea state represented in the Beaufort scale [0-12]. 0 - no waves at all, 12 - maximum sea state, very large waves.	**Min / Max** 0.0000 - 12.0000 **Default** 2.0000

6.4.13.25 Simulation / Environment / Default Weather / Global Water / Water Color

Parameter (type)	Values
Red **(float32)** The red component of the color.	**Min / Max** 0.0000 - 1.0000 **Default** 1.2000
Green **(float32)** The green component of the color.	**Min / Max** 0.0000 - 1.0000 **Default** 4.8000
Blue **(float32)** The blue component of the color.	**Min / Max** 0.0000 - 1.0000 **Default** 7.19900
Alpha **(float32)** The alpha channel value.	**Min / Max** 0.0000 - 1.0000 **Default** 0.88000

6.4.13.26 Simulation / Environment / Default Weather / Global Clouds / Cloud Layer 0

Parameter (type)	Values
Cloud Density (float32) Density of the cloud layer [0-1]. 0 = no clouds, 1 = full cloud coverage.	**Min / Max** 0.0000 - 1.0000 **Default** 1.0000
Cloud Coverage (float32) Describes the amount of coverage that should be applied within the density of the clouds [0-1]. 0 = No coverage, 1 = full coverage within the set density.	**Min / Max** 0.0000 - 1.0000 **Default** 0.6000
Cloud Type (float32) Type of the rendered clouds [0-1]. 0 = cumulus, 1 = stratus. Can interpolate between the two.	**Min / Max** 0.0000 - 1.0000 **Default** 0.18000
Base Elevation (float32) Altitude of the base (bottom) of the cloud layer.	**Default** 1100.0000
Bottom Transition Band Thickness (float32) The height of a vertical transition band below the cloud layer. This band produces a gradient.	**Default** 0.0000
Thickness (float32) The vertical thickness of the cloud layer.	**Default** 1500.0000
Top Transition Band Thickness (float32) The height of a vertical transition band above the cloud layer. This band produces a gradient.	**Default** 0.0000

6.4.13.27 Simulation / Environment / Default Weather / Global Clouds / Cloud Layer 1

Parameter (type)	Values
Cloud Density **(float32)** Density of the cloud layer [0-1]. 0 = no clouds, 1 = full cloud coverage.	**Min / Max** 0.0000 - 1.0000 **Default** 0.0000
Cloud Coverage **(float32)** Describes the amount of coverage that should be applied within the density of the clouds [0-1]. 0 = No coverage, 1 = full coverage within the set density.	**Min / Max** 0.0000 - 1.0000 **Default** 0.0000
Cloud Type **(float32)** Type of the rendered clouds [0-1]. 0 = cumulus, 1 = stratus. Can interpolate between the two.	**Min / Max** 0.0000 - 1.0000 **Default** 19000.0000
Base Elevation **(float32)** Altitude of the base (bottom) of the cloud layer.	**Default** 8200.0000
Bottom Transition Band Thickness **(float32)** The height of a vertical transition band below the cloud layer. This band produces a gradient.	**Default** 0.0000
Thickness **(float32)** The vertical thickness of the cloud layer.	**Default** 0.0000
Top Transition Band Thickness **(float32)** The height of a vertical transition band above the cloud layer. This band produces a gradient.	**Default** 0.0000

6.4.13.28 Simulation / Environment / Default Weather / Global Clouds / Cloud Layer 2

Parameter (type)	Values
Cloud Density **(float32)** Density of the cloud layer [0-1]. 0 = no clouds, 1 = full cloud coverage.	**Min / Max** 0.0000 - 1.0000 **Default** 0.0000
Cloud Coverage **(float32)** Describes the amount of coverage that should be applied within the density of the clouds [0-1]. 0 = No coverage, 1 = full coverage within the set density.	**Min / Max** 0.0000 - 1.0000 **Default** 1.0000
Cloud Type **(float32)** Type of the rendered clouds [0-1]. 0 = cumulus, 1 = stratus. Can interpolate between the two.	**Min / Max** 0.0000 - 1.0000 **Default** 0.0000
Base Elevation **(float32)** Altitude of the base (bottom) of the cloud layer.	**Default** 20000.0000
Bottom Transition Band Thickness **(float32)** The height of a vertical transition band below the cloud layer. This band produces a gradient.	**Default** 25.0000
Thickness **(float32)** The vertical thickness of the cloud layer.	**Default** 75.0000
Top Transition Band Thickness **(float32)** The height of a vertical transition band above the cloud layer. This band produces a gradient.	**Default** 100.0000

6.4.13.29 Simulation / Environment / Default Weather / Global Wind / Wind Speed

Parameter (type)	Values
Direction **(float64)** The wind direction from true north in degrees [0-360].	**Min / Max** 0.0000 - 360.0000 **Default** 0.0000
Horizontal Speed **(float64)** The horizontal wind speed parallel to the ellipsoid-tangential reference plane.	**Default** 0.0000
Vertical Speed **(float64)** The vertical wind speed. A positive value produces an updraft, while a negative value produces a downdraft.	**Default** 0.0000

6.4.13.30 Simulation / Environment / Default Weather / Lightning

Parameter (type)	Values
Random Lightning Spawning (boolean) Enables random lightning spawning.	**Default** false
Lightning Duration (float32) Determines how long a lightning strike lasts in seconds.	**Default** 0.001000
Lightning Spawn Interval (float32) The amount of time in-between lightning spawn attempts [seconds].	**Default** 0.5000
Lightning Spawn Chance (float32) The chance of a lightning bolt spawning during every lightning spawn interval [0-1].	**Default** 0.95000
Min Spawn Distance (float32) The minimum distance lightning can spawn from a view port [meters]. If there are multiple viewports, then a random viewport is selected to spawn lightning every spawn interval.	**Default** 2000.0000
Max Spawn Distance (float32) The maximum distance lightning can spawn from a view port [meters]. If there are multiple viewports, then a random viewport is selected to spawn lightning every spawn interval.	**Default** 35000.0000
Lightning Light Intensity (float32) Intensity of lightning point lights.	**Default** 2000.0000
Lightning Light Radius (float32) Radius of lightning point lights.	**Default** 5000.0000

6.4.13.31 Simulation / Environment / Default Rendering / Color Write Features

Parameter (type)	Values
Sky **(boolean)** Controls whether the skydome and atmosphere are enabled for rendering.	**Default** true
Sun **(boolean)** Controls whether the sun is enabled for rendering.	**Default** true
Moon **(boolean)** Controls whether the moon is enabled for rendering.	**Default** true
Stars **(boolean)** Controls whether the starfield is enabled for rendering.	**Default** true
Ground **(boolean)** Controls whether the ground surface is enabled for rendering.	**Default** true
Water **(boolean)** Controls whether water surfaces are enabled for rendering.	**Default** true
Biome Trees **(boolean)** Controls whether biome trees are enabled for rendering.	**Default** true
Biome Bushes **(boolean)** Controls whether biome bushes are enabled for rendering.	**Default** true
Biome Grass **(boolean)** Controls whether biome grass is enabled for rendering.	**Default** true
Geometry **(boolean)** Controls whether objects that are included in the terrain, such as buildings, are enabled for rendering.	**Default** true

Parameter (type)	Values
Point Cloud Data **(boolean)** Controls whether point cloud objects are enabled for rendering.	**Default** true
Objects **(boolean)** Controls whether dynamic objects, such as lifeform and platforms, are enabled for rendering.	**Default** true
Lights **(boolean)** Controls whether lights are enabled for rendering	**Default** true
Particles **(boolean)** Controls whether particles are enabled for rendering.	**Default** true
Clouds **(boolean)** Controls whether clouds are enabled for rendering.	**Default** true
Precipitation **(boolean)** Controls whether precipitation effects are enabled for rendering.	**Default** true

6.4.13.32 Simulation / Environment / Default Rendering / Depth Write Features

Parameter (type)	Values
Ground **(boolean)** Controls whether the ground surface is enabled for rendering to the depth buffer.	**Default** true
Water **(boolean)** Controls whether water surfaces are enabled for rendering to the depth buffer.	**Default** true
Biome Trees **(boolean)** Controls whether biome trees are enabled for rendering to the depth buffer.	**Default** true
Biome Bushes **(boolean)** Controls whether biome bushes are enabled for rendering to the depth buffer.	**Default** true
Biome Grass **(boolean)** Controls whether biome grass is enabled for rendering to the depth buffer.	**Default** true
Geometry **(boolean)** Controls whether objects that are included in the terrain, such as buildings, are enabled for rendering to the depth buffer.	**Default** true
Point Cloud Data **(boolean)** Controls whether point cloud objects are enabled for rendering to the depth buffer.	**Default** true
Starfield Intensity **(float32)** The intensity of the starfield. [0-1].	**Min /Max** 0.0000 - 1.0000 **Default** 1.0000

6.4.13.33 Simulation / Terrain Deformation

Parameter (type)	Values
Craters **(boolean)** Enable creation of craters.	**Default** true
Soft Surface Deformations **(boolean)** Enable deformation of soft surfaces such as snow.	**Default** false

6.4.13.34 Simulation / Laser / Default Attributes

Parameter (type)	Values
Wavelength **(uint32)** The wavelength of the laser in nanometers.	**Default** 510
Intensity **(uint32)** The intensity of the laser in milliwatts.	**Default** 1000
Lasers Count **(uint32)** Number of lasers to be drawn for a laser instance (a laser instance can be composed of many lasers so that divergence can be rendered).	**Default** 24
Max Length **(uint32)** Maximum laser length in meters.	**Default** 43000

6.4.13.35 Simulation / Laser / Intersection

Parameter (type)	Values
Min Detail **(uint8)** The minimum detail for laser intersections. Pass 0 to use the default settings specified in BlueController.	**Default** 0
Max Detail **(uint8)** The maximum detail for laser intersections. Pass 0 to use the default settings specified in BlueController.	**Default** 15
Laser Intersection Time **(float32)** If true, lasers will do object intersection tests	**Default** 0.5000
Laser Intersection Enabled **(boolean)** If true, lasers will do object intersection tests	**Default** true

6.4.13.36 Simulation / Ropes / Rope System

Rope system related settings

6.4.13.37 Simulation / Ropes / Rope System / Implicit Color

Implicit color of ropes. Applied to ropes that do not have color explicitly specified.

Parameter (type)	Values
Red **(float32)** The red component of the color.	**Min / Max** 0.0000 - 1.0000 **Default** 0.0000
Green **(float32)** The green component of the color.	**Min / Max** 0.0000 - 1.0000 **Default** 0.0000
Blue **(float32)** The blue component of the color.	**Min / Max** 0.0000 - 1.0000 **Default** 0.0000
Alpha **(float32)** The alpha channel value.	**Min / Max** 0.0000 - 1.0000 **Default** 0.0000
Implicit Thickness **(float32)** Implicit thickness of ropes in meters. Applied to ropes that do not have thickness explicitly specified.	**Min / Max** 0.0000 - 1.0000 **Default** 0.0000
Implicit Segments Count **(uint8)** Implicit segments count of ropes. Applied to ropes that do not have segments count explicitly specified.	**Min / Max** 1 - UINT8_MAX **Default** 16
Bezier Curve Method **(enum)** Method for approximating Bezier curves.	• DeCasteljau • DeCasteljauApproximation • BernsteinPolynomials **Default** BernsteinPolynomials

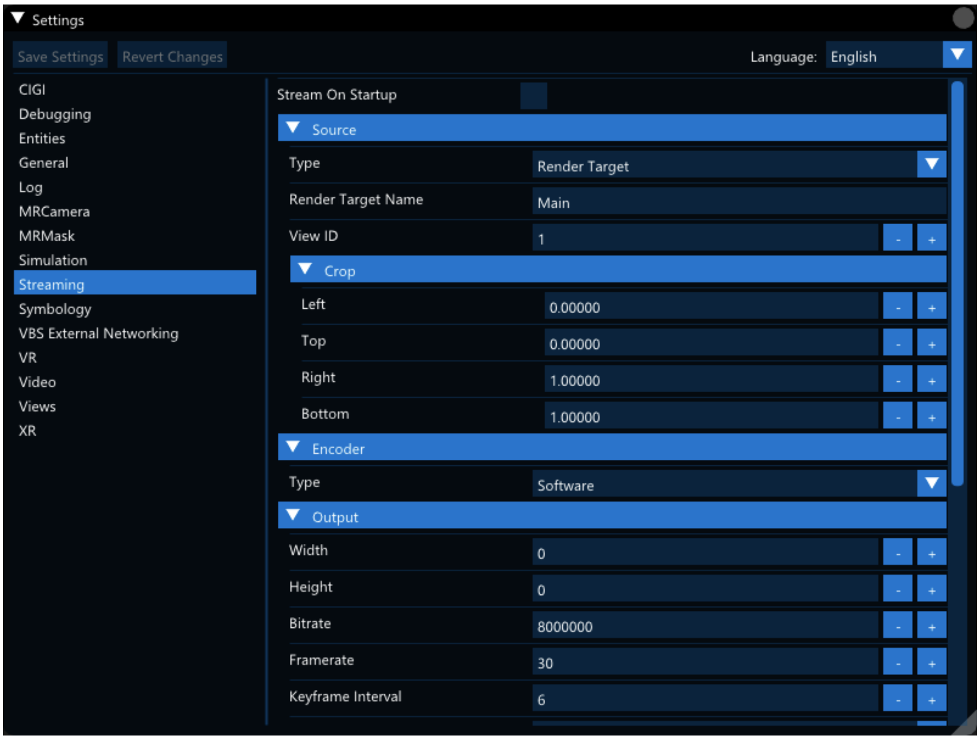
6.4.14 Streaming Settings

VBS Blue IG contains a component for video streaming (STANAG-4586 compliant H.264 in MPEG-2 TS container).

Use the Debug UI Video Streaming (on page 415) window to specify which render target or view to be streamed.

These video streams are further customized using either of the following methods:

- **Settings UI** - Press **Tab** to open the VBS Blue IG Settings (on page 201) and make adjustments, as required.



- **Edit XML file** - Open the `Streaming.xml` file, located in the following directory and edit, as required:

  `%LOCALAPPDATA%\Bohemia Interactive Simulations\`
  `VBS Blue IG\`*`version`*`\Settings\`

The following table describes the type and purpose for each setting.

6.4.14.1 Streaming

Parameter (type)	Values
Stream On Startup **(boolean)** If true, the stream will start right when the program starts. If false, the user will need to start the stream manually at runtime.	**Default** false

6.4.14.2 Streaming / Source

Parameter (type)	Values
Type **(enum)** The source from which the video will be streamed.	• View • Render Target **Default** Render Target
Render Target Name **(string)** The name of the render target which will be streamed and of which screenshots will be taken. "Main" identifies the main window. When empty, SourceView is used instead.	**Default** Main
View ID **(uint32)** The ID of the view which will be streamed and of which screenshots will be taken. Ignored when SourceRenderTarget is a non-empty string.	**Min / Max** 0 - UINT32_MAX **Default** 1

6.4.14.3 Streaming / Source / Crop

Parameter (type)	Values
Left **(float32)** The normalized offset of the left edge of the video stream, from the left edge of the camera viewport or render target. Range [0-Right).	**Min / Max** 0.0 - 1.0 **Default** 0.0
Top **(float32)** The normalized offset of the top edge of the video stream, from the top edge of the camera viewport or render target. Range [0-Bottom).	**Min / Max** 0.0 - 1.0 **Default** 0.0
Right **(float32)** The normalized offset of the right edge of the video stream, from the left edge of the camera viewport or render target. Range (Left-1].	**Min / Max** 0.0 - 1.0 **Default** 1.0
Bottom **(float32)** The normalized offset of the bottom edge of the video stream, from the top edge of the camera viewport or render target. Range (Top-1].	**Min / Max** 0.0 - 1.0 **Default** 1.0

6.4.14.4 Streaming / Encoder

Parameter (type)	Values
Type **(enum)** Use NVENC (Nvidia Encoder) to offload the video encoding to the GPU.	• Software NVENC **Default** Software

6.4.14.5 Streaming / Output

Parameter (type)	Values
Width **(uint32)** The width the video will be streamed at, where 0 means current view/render target width.	**Min / Max** 0 - UINT32_MAX **Default** 0
Height **(uint32)** The height the video will be streamed at, where 0 means current view/render target height .	**Min / Max** 0 - UINT32_MAX **Default** **768**
Bitrate **(uint32)** The bitrate the video will be streamed at.	**Min / Max** 0 - UINT32_MAX **Default** 8000000
Framerate **(uint16)** Target frame rate of the output video in frames per second (FPS).	**Min / Max** 1 - 240 **Default** 30
Keyframe Interval **(uint8)** Keyframe interval of the output video specified by number of frames. Making it too short might increase network traffic significantly.	**Min / Max** 1 - UINT32_MAX **Default** 6
Type **(enum)** Streaming output type.	File UDP HTTP **Default** UDP

6.4.14.6 Streaming / Output / File

Parameter (type)	Values
File Name **(string)** The filename that the video will be saved to.	%userprofile%\Videos\VBS Blue IG\Stream.mp4
Options **(string)** File options.	**Default** (EmptyString)

6.4.14.7 Streaming / Output / UDP

Parameter (type)	Values
URL **(string)** The multicast address the video will be streamed to.	udp://224.1.1.1:5000
Options **(string)** The options that will be appended onto the stream. If streaming to VLC 3.0.1, pkt_size=1316 must be set.	pkt_size=1316
Send Interface **(string)** The interface that will be appended to the stream options to specify a NIC.	**Default** (EmptyString)

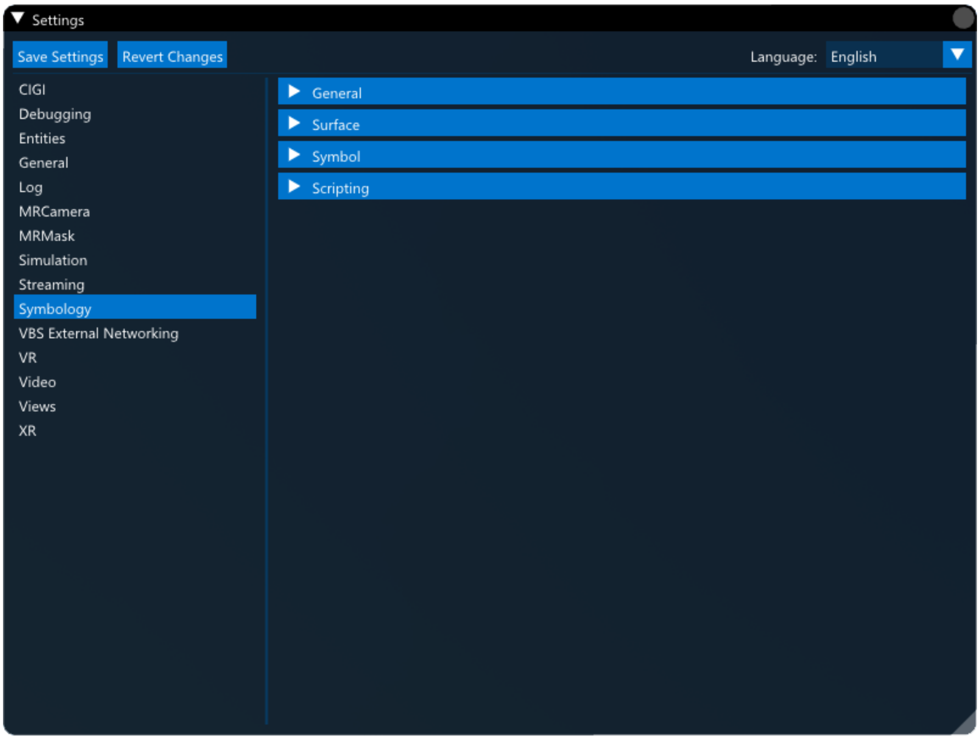
6.4.14.8 Streaming / Output / HTTP

Parameter (type)	Values
HTTP Server Port **(uint16)** Port used by the http video stream.	**Min / Max** 0 - UINT16_MAX **Default** 11235
Options **(string)** HTTP options.	**Default** (EmptyString)

6.4.15 Symbology Settings

Customize General settings for Symbology using either of the following methods:

- **Settings UI** - Press **Tab** to open the VBS Blue IG Settings (on page 201) and make adjustments, as required.



- **Edit XML file** - Open the `Symbology.xml` file, located in the following directory and edit, as required:

  `%LOCALAPPDATA%\Bohemia Interactive Simulations\`
  `VBS Blue IG\`*`version`*`\Settings\`

The tables below describe the type and purpose for each setting.

6.4.15.1 Symbology / General

Parameter (type)	Values
Max Update Threads **(uint32)** The maximum number of threads to use for updating Symbology, per Symbology context (0 = Number of logical CPU cores)	**Default** 0
Blending In Linear Gamma **(boolean)** If enabled, then alpha blending will occur in linear gamma space (Gamma = 1.0). Blending in linear gamma is colorimetricly correct, however most applications do not use this for blending, and instead blend in SRGB gamma space (Gamma = ~2.2).	**Default** false

6.4.15.2 Symbology / Surface

Parameter (type)	Values
Default Multi-Sampling **(uint8)** Set the default multi-sampling state on all surfaces. Value must be between 0 (disabled) and D3D11_MAX_MULTISAMPLE_SAMPLE_COUNT (32) that your graphics card supports.	**Min / Max** 0 -32 **Default** 0
Default Texture Filter **(enum)** Set the default texture filter method on all surfaces.	• Point • Linear • Anisotropic **Default** Anisotropic

6.4.15.3 Symbology / Symbol / Browser

Parameter (type)	Values
Client Domains Directory **(string)** The path for the `client://` scheme domains directory for the embedded web browser. The path may be specified as a relative path to the SymbologyComponent component directory.	**Default** Domains
JavaScript App Object Name **(string)** With Browser symbols, the variable name of the global JavaScript object variable for calling native functions.	**Default** app
Remote Debugging Port **(uint16)** With Browser symbols, set to a value between 1024 and 65535 to enable remote debugging on the specific port. The embedded web browser can be remotely debugged via a Chrome browser window, navigating to http://localhost:(port) Set to 0 to disable remote debugging.	**Default** 9000
Enable GPU **(boolean)** With Browser symbols, set to true to enable GPU rendering (i.e. enable WebGL support).	**Default** true

6.4.15.4 Symbology / Symbol / Polygon

Parameter (type)	Values
Default Line Scale Thickness (boolean) With Polygon Line symbols, if the line thickness is scaled when the symbol is scaled, by default.	**Default** 1
Default Line Joint Style (enum) With Polygon Line symbols, the default joint style used to connect line strips and line loops.	• None • Mitered • Beveled • Rounded **Default** None
Default Line Start Cap Style (enum) With Polygon Line symbols, the default start cap style used to start lines and line strips.	• Butt • Round • Square • Triangle **Default** Butt
Default Line End Cap Style (enum) With Polygon Line symbols, the default end cap style used to end lines and line strips.	• Butt • Round • Square • Triangle **Default** Butt
Default Triangle Edge Smoothing (boolean) With Polygon Triangle symbols, the default state of edge smoothing. Edge smoothing is a method of anti-aliasing, without the performance issues of multi-sampling. Edge smoothing should not be used on triangles with high vertex counts.	**Default** false

6.4.15.5 Symbology / Symbol / SVG

Parameter (type)	Values
Default Line Scaling **(boolean)** With SVG symbols, if the SVG path line widths are scaled when the symbol is scaled, by default.	**Default** true
Default Outline Mode **(boolean)** With SVG symbols, if outline mode is enabled, by default. Outline mode forces all SVG paths to have a line width as per Outline Thickness	**Default** false
Default Line Thickness **(float32)** With SVG symbols, the default line thickness when using outline mode.	**Default** 1.0000

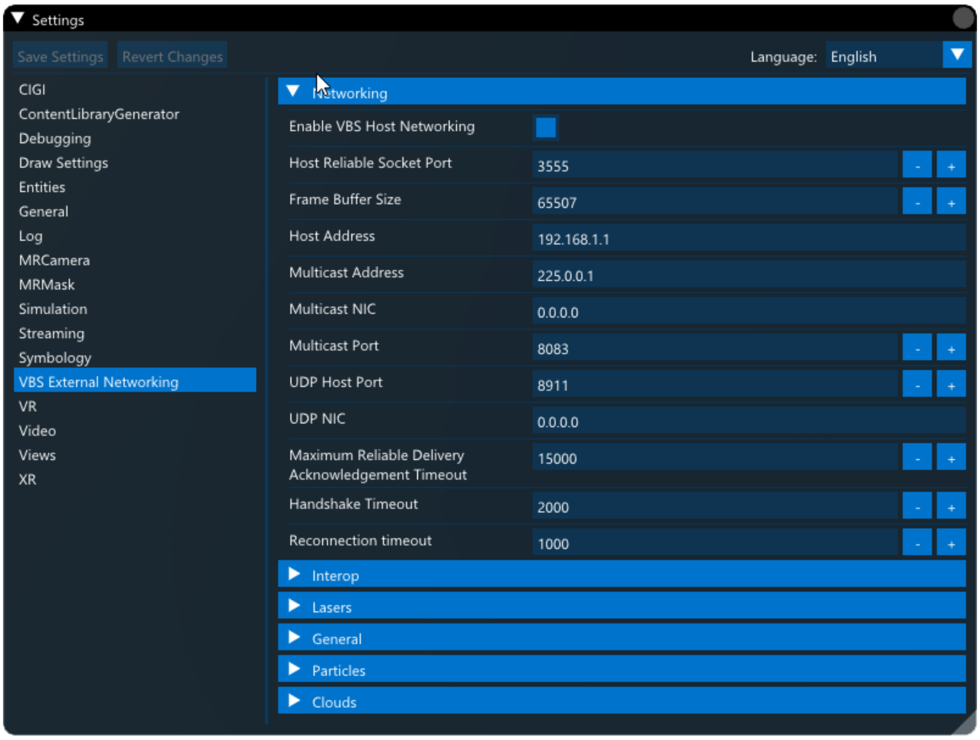
6.4.15.6 Symbology / Scripting

Parameter (type)	Values
Symbology Lua Scripts **(array)** A list of all Symbology scripts that will be executed upon application start.	**Default** empty

6.4.16 VBS External Networking Settings

Customize settings related to using VBS as a host using either of the following methods:

- **Settings UI** - Press **Tab** to open the VBS Blue IG Settings (on page 201) and make adjustments, as required.



- **Edit XML file** - Open the `VBSExternalNetworking.xml` file, located in the following directory and edit, as required:

  `%LOCALAPPDATA%\Bohemia Interactive Simulations\`
  `VBS Blue IG\`*`version`*`\Settings\`

The tables below describe the type and purpose for each setting.

6.4.16.1 VBS External Networking / Networking

Parameter (type)	Values
Enable VBS Host Networking (boolean) Enables / disables VBS Host Networking. The setting can be overridden using `-vbsHostNet` command line parameter.	**Default** true
Host Reliable Socket Port (uint16) Port of the reliable socket for the host. This port is used to attempt a connection with the host.	**Default** 3555
Frame Buffer Size (uint16) Max size a single network frame can be.The largest this value can be is 65507 which is UDP datagram max. https://en.wikipedia.org/wiki/User_Datagram_Protocol.	**Min / Max** 1024 - 65507 **Default** 65507
Host Address (string) Address of the host. This address is used to attempt a connection with the host.	**Default** 127.0.0.1
Multicast Address (string) The address of the multicast group used to send UDP datagrams from the host to IG clients.	**Default** 225.0.0.1
Multicast NIC (string) The address of the network adapter used to connect to the multicast group specified by Multicast Address. '0.0.0.0' to use the default adapter.	**Default** 0.0.0.0
Multicast Port (uint16) The port of the multicast group used to send UDP datagrams from the host to IG clients.	**Default** 8083
UDP Host Port (uint16) UDP port of the host. This port is used to send UDP datagrams to the host.	**Default** 8911

Parameter (type)	Values
UDP NIC **(string)** The address of the network adapter to use for sending UDP datagrams to the host. '0.0.0.0' to use the default adapter.	**Default** 0.0.0.0
Maximum Reliable Delivery Acknowledgment Timeout **(uint32)** Maximum timeout (in milliseconds) for the reliable data delivery to be acknowledged by the receiving host. If the timeout is exceeded, the connection with the host is dropped.	1 - UINT32_MAX **Default:** 15000
Handshake Timeout **(uint32)** Number of milliseconds to wait for a host to deliver a connection request and for a client to respond to it before timing out	**Default** 2000
Reconnection timeout **(uint32)** Number of milliseconds to wait before another attempt to reconnect to a host is made	**Default** 1000

6.4.16.2 VBS External Networking / Interop

Parameter (type)	Values
Craters Enabled **(boolean)** If True, craters will be created only if their radius is greater than or equal to MinimumCraterRadiusToCreate. If False, no craters will be created.	**Default** true
Minimum Crater Radius For 3D Craters **(float64)** The minimum radius in meters that a crater must be in order for it to be created as 3D. Craters with a smaller radius will be created as flat 2D textures, which are less performance intensive. When set to 0, all craters will be created as 3D. Ignored if Craters Enabled is set to False.	**Default** 0.20000
3D Crater Depth **(float32)** Depth scale of 3D craters within the range of [0,1], where 0 creates a crater with just a 2D decal texture and no terrain deformation, and 1 creates a crater with both the decal texture and a default amount of terrain deformation based on its radius. Only applicable to craters classified as 3D by the Minimum Crater Radius For 3D Craters setting. Ignored if Craters Enabled is set to False.	**Default** 1.00000
Enable Soft Surface Deformations **(boolean)** If true, enables soft surface (e.g. snow) deformations to be created.	**Default** true
Interpolation Enabled **(boolean)**	**Default** true
Particles Enabled **(boolean)** If True, Scripted / Drop Particles will be simulated. If False, no Scripted / Drop particles will be created.	**Default** true

6.4.16.3 VBS External Networking / Interop / Weather

Parameter (type)	Values
Fog Enabled **(boolean)** Toggles interop fog updates.	**Default** true
Precipitation Enabled **(boolean)** Toggles interop precipitation updates.	**Default** true
Precipitation Min Altitude **(float32)** The minimum value for precipitation altitude, in meters.	**Default** 500.0000
Precipitation Max Altitude **(float32)** The maximum value for precipitation altitude, in meters.	**Default** 20000.0000
Wind Enabled **(boolean)** Toggles interop wind updates.	**Default** true

6.4.16.4 VBS External Networking / Interop / Clamping / VBS3 Clamping Settings / Vehicle

Parameter (type)	Values
Ground Vehicle Clamp Mode (enum)	• None • Height • Normal • Height And Normal **Default** Height And Normal
Air Vehicle Clamp Mode (enum)	• None • Height • Normal • Height And Normal **Default** None
Water Vehicle Clamp Mode (enum)	• None • Height • Normal • HeightAndNormal **Default** None

6.4.16.5 VBS External Networking / Interop / Clamping / VBS3 Clamping Settings / Lifeform

Parameter (type)	Values
Lifeform Standing Clamp Mode (enum)	• None • Height • Normal • Height And Normal **Default** Height
Lifeform Crouching Clamp Mode (enum)	• None • Height • Normal • Height And Normal **Default** Height
Lifeform Prone Clamp Mode (enum)	• None • Height • Normal • Height And Normal **Default** Height And Normal
Lifeform Dead Clamp Mode (enum)	• None • Height • Normal • Height And Normal **Default** Height And Normal

6.4.16.6 VBS External Networking / Interop / Clamping / VBS3 Clamping Settings / Object

Parameter (type)	Values
Static Object Clamp Mode **(enum)**	• None • Height • Normal • Height And Normal Default Height And Normal
Structures Clamp Mode **(enum)**	• None • Height • Normal • Height And Normal **Default** Height
Ignored Bones **(string[])** List of bones to ignore when a vehicle is clamped	**Default:** See Ignored Bones Values (on the next page) below.
Height Above Surface Enabled **(boolean)** If True, the height above the underlying surface will be applied as an offset for entities. If False, no offset will be applied and entities will be clamped exactly to the surface. This setting must be the same value on the Host and the IG for correct operation.	**Default** true
Object Clamping Enabled **(boolean)** If True, entities will use intersection testing to clamp to underlying object surfaces. If False, entities will only clamp to ground or water. This setting must be the same value on the Host and the IG for correct operation.	**Default** true

Ignored Bones Values

Wheel_1_1_damper	Wheel_6_1_damper	Wheel_3_1	Wheel_8_1
Wheel_1_2_damper	Wheel_6_2_damper	Wheel_3_2	Wheel_8_2
Wheel_2_1_damper	Wheel_7_1_damper	Wheel_4_1	Wheel_9_1
Wheel_2_2_damper	Wheel_7_2_damper	Wheel_4_2	Wheel_9_2
Wheel_3_1_damper	Wheel_8_1_damper	Wheel_5_1	wheel_roller_1_1
Wheel_3_2_damper	Wheel_8_2_damper	Wheel_5_2	wheel_roller_1_2
Wheel_4_1_damper	Wheel_1_1	Wheel_6_1	wheel_roller_2_1
Wheel_4_2_damper	Wheel_1_2	Wheel_6_2	wheel_roller_2_2
Wheel_5_1_damper	Wheel_2_1	Wheel_7_1	wheel_roller_3_1
Wheel_5_2_damper	Wheel_2_2	Wheel_7_2	wheel_roller_3_2

6.4.16.7 VBS External Networking / Interop / Clamping / VBS4 Clamping Settings / Vehicle

Parameter (type)	Values
Ground Vehicle Clamp Mode (enum)	• None • Height • Normal • Height And Normal **Default** None
Air Vehicle Clamp Mode (enum)	• None • Height • Normal • Height And Normal **Default** None
Water Vehicle Clamp Mode (enum)	• None • Height • Normal • Height And Normal **Default** None

6.4.16.8 VBS External Networking / Interop / Clamping / VBS4 Clamping Settings / Lifeform

Parameter (type)	Values
Lifeform Standing Clamp Mode (enum)	• None • Height • Normal • Height And Normal **Default** None
Lifeform Crouching Clamp Mode (enum)	• None • Height • Normal • Height And Normal **Default** None
Lifeform Prone Clamp Mode (enum)	• None • Height • Normal • Height And Normal **Default** None
Lifeform Dead Clamp Mode (enum)	• None • Height • Normal • Height And Normal **Default** None

6.4.16.9 VBS External Networking / Interop / Clamping / VBS4 Clamping Settings / Object

Parameter (type)	Values
Static Object Clamp Mode (enum)	• None • Height • Normal • HeightAndNormal **Default** None
Structures Clamp Mode (enum)	• None • Height • Normal • HeightAndNormal **Default** None
Ignored Bones (string[]) List of bones to ignore when a vehicle is clamped	**Default:** See Ignored Bones Values (on the next page) below.
Height Above Surface Enabled (boolean) If True, the height above the underlying surface will be applied as an offset for entities. If False, no offset will be applied and entities will be clamped exactly to the surface. This setting must be the same value on the Host and the IG for correct operation.	**Default** false
Object Clamping Enabled (boolean) If True, entities will use intersection testing to clamp to underlying object surfaces. If False, entities will only clamp to ground or water. This setting must be the same value on the Host and the IG for correct operation.	**Default** false

Ignored Bones Values

Wheel_1_1_damper	Wheel_6_1_damper	Wheel_3_1	Wheel_8_1
Wheel_1_2_damper	Wheel_6_2_damper	Wheel_3_2	Wheel_8_2
Wheel_2_1_damper	Wheel_7_1_damper	Wheel_4_1	Wheel_9_1
Wheel_2_2_damper	Wheel_7_2_damper	Wheel_4_2	Wheel_9_2
Wheel_3_1_damper	Wheel_8_1_damper	Wheel_5_1	wheel_roller_1_1
Wheel_3_2_damper	Wheel_8_2_damper	Wheel_5_2	wheel_roller_1_2
Wheel_4_1_damper	Wheel_1_1	Wheel_6_1	wheel_roller_2_1
Wheel_4_2_damper	Wheel_1_2	Wheel_6_2	wheel_roller_2_2
Wheel_5_1_damper	Wheel_2_1	Wheel_7_1	wheel_roller_3_1
Wheel_5_2_damper	Wheel_2_2	Wheel_7_2	wheel_roller_3_2

6.4.16.10 VBS External Networking / Interop / Client / Blacklist

Parameter (type)	Values
File **(string)**	**Default** entities.blacklist
Non-Smoothed Types **(string[])** Class names that will not have their position be smoothed by interpolation (if enabled).	**Default** • bisim_p1sart_rescue_basket_x • bisim_p1sart_rescue_strop_x • bisim_p1sart_rescue_litter_x • bisim_p1sart_rescue_seasoning_weight_x

6.4.16.11 VBS External Networking / Interop / AAR

Parameter (type)	Values
Bone Request Threshold **(float32)** Distance in meters from a view before lifeforms stop requesting bone data.	**Default** 500.0000
Bone Request Minimum Frequency **(uint32)** Minimum frequency in frames to request bone data when a lifeform is within BoneRequestThreshold.	**Default** 1
Bone Request Distance Modifier **(float32)** Distance modifier used to calculate bone request frequency. This value is multiplied by the distance.	**Default** 1.50000

6.4.16.12 VBS External Networking / Interop / Vehicle Tracks

Parameter (type)	Values
Tracks State **(enum)** Enable state and behavior of the tracks generation for vehicles.	• Use Global Settings • Auto • Forced off **Default** Use Global Settings
Dust Trails State **(enum)** Enable state and behavior of the dust trails generation for vehicles.	• Use Global Settings • Auto • Forced off **Default** Forced off

6.4.16.13 VBS External Networking / Interop / Lifeform Tracks

Parameter (type)	Values
Tracks State **(enum)** Enable state and behavior of the tracks generation for lifeforms.	• Use Global Settings • Auto • Forced off **Default** Use Global Settings
Snow Trails State **(enum)** Initial enable state of the snow trails generation for lifeforms.	• Use Global Settings • Auto • Forced off **Default** Forced off

6.4.16.14 VBS External Networking / Lasers

Parameter (type)	Values
Default Wavelength **(uint32)** The default wavelength in nanometers to use when creating a laser.	**Default** 781

Parameter (type)	Values
Use Laser Endpoint From VBS (boolean) If true, use the laser end point provided by VBS. If false, calculate the laser end point using an intersection test in VBS Blue IG.	**Default** true
Laser Animation Distance (float32) The default distance from laser startpoint to force the character bounding box to be. This is to ensure the character can aim correctly in a multi view setup. Range 0-10000.	**Min / Max** 0 - 10000.0000 **Default** 1000.0000

6.4.16.15 VBS External Networking / General

Parameter (type)	Values
Enable Extra Rotor Effects (boolean) If enabled, helicopters will generate rotor wash effects when near the terrain.	**Default** false
Host-Referenced Terrain Object Search Radius (float32) Radius in meters within which terrain objects referenced by the host are searched, for example, to apply building destruction or animations. Larger distance may be necessary to account for terrain differences between the host and the IG.	**Min / Max** 0.001000 - 100.0000 **Default** 1.0000
Enable Enable Performance Metrics Logging On Scenario Start (boolean) If enabled, VBS Blue IG will automatically start logging performance metrics when the scenario is started by the VBS Host. The performance logs will be saved to `<Product Directory>\PerformanceLogs` with a file name matching the scenario name. The metrics to be logged are controlled by the Debugging / Metrics / Log to File settings.	**Default** false

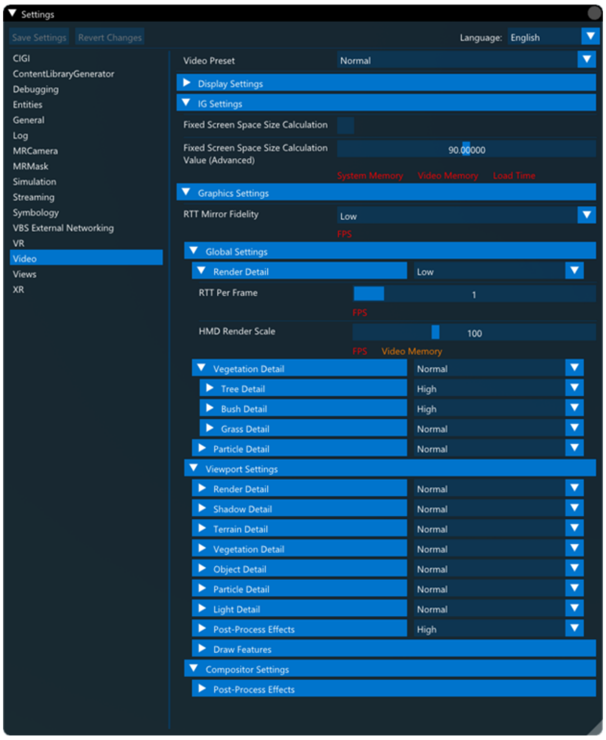
6.4.16.16 VBS External Networking / Particles

Parameter (type)	Values
Alpha Cull **(float32)** The minimum threshold that a particle alpha color must exceed in order to be created in the scene. Minimum 0.0f, Maximum 1.0f.	**Min / Max** 0.0000 - 1.0000 **Default** 0.00000
Lifetime Modifier **(float32)** Multiplicative modifier applied to all created particle sources lifetimes. The higher the value, the longer the particles last. Minimum 0.01f, Maximum 100.0f.	**Min / Max** 0.01000 - 100.0000 **Default** 1.00000
Spawntime Modifier **(float32)** Multiplicative modifier applied to all created particle sources spawn timers. The higher the value, the less overall particles spawned. Minimum 0.01f, Maximum 100.0f.	**Min / Max** 0.010000 - 100.0000 **Default** 1.0000

6.4.17 Video Settings

The Video Settings menu can be used to modify graphics quality settings for VBS Blue IG clients in real-time using either of the following methods:

- **Settings UI** - Press **Tab** to open the VBS Blue IG Settings (on page 201) and make adjustments, as required.



- **Edit XML file** - Open the `VideoSettings.xml` file, located in the following directory and edit, as required:

  `%LOCALAPPDATA%\Bohemia Interactive Simulations\`
  `VBS Blue IG\`*`version`*`\Settings\`

Follow these steps:

1. Press **Tab** to open the Settings UI on any VBS Blue IG client.
2. Video settings are grouped into aggregates. Configure settings by choosing their respective preset values (usually **low**, **normal**, **high**, **ultra**).
3. For finer control over visual output, users can still manipulate each setting within an aggregate on an individual level.
4. Click **Save Settings**.
5. Restart VBS Blue IG.

The tables below describe the type and purpose for each setting.

6.4.17.1 Video / Display Settings

Parameter (type)	Values
Window Position X **(int32)** Position of the main window in the X axis.	**Min / Max** -1080 - INT32_MAX **Default** 100
Window Position Y **(int32)** Position of the main window in the Y axis.	**Min / Max** 1080 - INT32_MAX **Default** 100
Window Width **(int32)** Size of the main window in the X axis.	**Min / Max** 64 - INT32_MAX **Default** 1920
Window Height **(int32)** Size of the main window in the Y axis.	**Min / Max** 48 - INT32_MAX **Default** 1080
Borderless Window **(boolean)** When enabled the standard window borders of the application will be removed.	**Default** false
Top-most Window **(boolean)** When enabled the application will bring itself to the forefront of all other windows.	**Default** false
Fullscreen **(boolean)** When enabled the application will switch to fullscreen.	**Default** false
VSync **(boolean)** Display's vertical synchronization. This option will limit the frame rate to the refresh rate, and may drastically reduce FPS. However, it is recommended to have this enabled to prevent screen tearing.	**Default** true

Parameter (type)	Values
Brightness (float32) The brightness of the image.	**Min / Max** 0.5000 - 1.5000 **Default** 1.0000
Gamma (float32) Exponent of output image gamma. 2.4 is sRGB gamma.	**Min / Max** 1.0000 - 3.50000 **Default** 2.40000

6.4.17.2 Video/ IG Settings

Parameter (type)	Values
Fixed Screen Space Size Calculation (boolean) Enable to use a fixed field of view for options that are otherwise affected by field of view, such as Scene Detail and other draw distance based options. This option should be enabled when using a multichannel configuration, to ensure that different views with different field of views have identical draw distances. Disable this option for use cases that involve dynamic zooming, such as a UAV camera, that would need to have an increased draw distance at higher zoom levels.	**Default** false
Fixed Screen Space Size Calculation Value (Advanced) (float32) Fixed field of view value in degrees to use when the Fixed Screen Space Size Calculation option is enabled. When the option is disabled, this has no effect. This an advanced setting which should not need to be changed from the default of 90 in most cases. Setting the value too low compared to the actual field of view may result in excessive RAM and VRAM consumption, leading to system instability.	**Min / Max** 0.01 - 179.99 **Default** 90.000

6.4.17.3 Video/ Graphics Settings

Parameter (type)	Values
RTT Mirror Fidelity **(enum)** Controls the quality and overall fidelity of vehicle mirrors.	• Off • Low • Normal • High • Ultra • 4K **Default** Low

6.4.17.4 Video / Graphics Settings / Global Settings / Render Detail

Parameter (type)	Values
RTT Per Frame **(int32)** How many dynamic render-to-texture resources are rendered in one frame.	**Min / Max** 1 - 8 **Default** 1
HMD Render Scale **(int32)** Supersampling setting for HMD. Applied on top of any supersampling configured in the HMD vendor settings (e.g. Oculus pixel per display).	**Min / Max** 50 - 200 **Default** 100

6.4.17.5 Video / Graphics Settings / Global Settings / Vegetation Detail / Tree Detail

Parameter (type)	Values
Tree Types Limit (enum) Determines how many tree types there is going to be per surface in the scene. The less vegetation types, the better the performance.	• 1 (one type per surface) • 2 (two types per surface) • 3 (three types per surface) • 4 (four types per surface) • 5 (five types per surface) • Unlimited **Default** Unlimited

6.4.17.6 Video / Graphics Settings / Global Settings / Vegetation Detail / Bush Detail

Parameter (type)	Values
Bush Types Limit (enum) Determines how many tree types there is going to be per surface in the scene. The less vegetation types, the better the performance.	• 1 (one type per surface) • 2 (two types per surface) • 3 (three types per surface) • 4 (four types per surface) • 5 (five types per surface) • Unlimited **Default** Unlimited

6.4.17.7 Video / Graphics Settings / Global Settings / Vegetation Detail / Grass Detail

Parameter (type)	Values
Grass Types Limit **(enum)** Determines how many grass types there is going to be per surface in the scene. The less vegetation types, the better the performance.	• 1 (one type per surface) • 2 (two types per surface) • 3 (three types per surface) • 4 (four types per surface) • 5 (five types per surface) • Unlimited **Default** 2 (two types per surface)

6.4.17.8 Video / Graphics Settings / Global Settings / Particle Detail

Parameter (type)	Values
Particle Count Limit **(int32)** Number of simulated particles in the scene is limited by this number.	**Min / Max** 100 - 20000 **Default** 5000

6.4.17.9 Video / Graphics Settings / Viewport Settings / Render Detail

Parameter (type)	Values
DLSS **(enum)** Deep Learning Super Sampling. Available only on Nvidia GPUs with DLSS support. Enabling this feature will disable Render Resolution and FSAA for the given window.	• Disabled • Ultra-Performance • Performance • Balanced • Quality • Ultra-Quality **Default** Quality
Render Resolution **(int32)** Render resolution of viewports. Values higher than 100% performs supersampling of the viewport. Values lower than 100% will result in a blurry, less detailed viewport.	**Min / Max** 10 - 400 **Default** 100
MSAA **(enum)** Level of multisampling used in the scene. This will reduce the jagged edges of geometry with higher values, at the cost of performance and video memory. **NOTE** This setting is not used when DLSS is enabled.	2x MSAA 4x MSAA 8x MSAA **Default:** 4x MSAA
Transparency Render Scale **(int32)** Render resolution of transparent objects in the exterior scene. Lower setup will result in more blurry transparent objects, but faster rendering.	**Min / Max** 25 - 100 **Default** 100
Transparency Method **(enum)** Method of rendering transparent objects. Simple method just renders transparent exterior objects in lower resolution. MultiResolution method combines low resolution rendering with full resolution rendering.	• Simple • MultiResolution • MultiDistance **Default** Simple

Parameter (type)	Values
Multi-Projection Technology (enum) Toggles use of Nvidia VRWorks single-pass stereo or multi-view rendering for accelerating HMD and multi-projection rendering. Requires an Nvidia GPU with SPS or MVR support.	**Min / Max** • None • SPS • MVR / SPS **Default** MVR / SPS
Anisotropy Quality (int32) Level of anisotropic filtering used in the scene. This will improve the detail of textures when viewed at sharp angles.	**Min / Max** 1 - 16 **Default** 16

6.4.17.10 Video / Graphics Settings / Viewport Settings / Shadow Detail

Parameter (type)	Values
Biome Shadows (boolean) Enables / disables shadows cast by biome vegetation.	**Default** true
Exterior Shadow Cascades (int32) Controls the number of exterior shadow maps covering the view: 2, 3, 4. More cascades, better shadow quality.	**Min / Max** 2 - 4 **Default** 3
Interior Shadow Cascades (int32) Controls the number of interior shadow maps covering the view: 1, 2. More cascades, better shadow quality.	**Min / Max** 1 - 2 **Default** 2
Exterior Shadow Quality (int32) Controls the exterior shadow map resolution: 0 - exterior shadows disabled, 1 - low, 2 - good, 3 - ultra.	**Min / Max** 0 - 3 **Default** 2
Interior Shadow Quality (int32) Controls the interior shadow map resolution: 0 - interior shadows disabled, 1 - low, 2 - good, 3 - ultra.	**Min / Max** 0 - 3 **Default** 2

Parameter (type)	Values
Exterior Shadow Draw Distance (float32) Controls draw distance of exterior shadows in meters. Larger draw distance will result in worse shadow quality and worse performance.	**Min / Max** 200 - 2000 **Default** 750

6.4.17.11 Video / Graphics Settings / Viewport Settings / Terrain Detail

Parameter (type)	Values
Segment Subdivision Depth (int32) Limits the maximum terrain segment level in LOD. A value of 23 is the highest quality LOD for terrain. The lower the value, the lower the ground detail.	**Min / Max** 0 - 23 **Default** 20
Terrain Vertex Count (enum) The higher the vertex count the more detailed the terrain mesh is. This is especially visible on distance hills / mountains.	• 17 • 33 • 65 **Default** 33
Terrain Detail (float32) Draw distance of terrain. Value is distance in meters up to which a segment with one meter size is split into smaller segments when camera's field of view is 90 degrees. Segment with k-meter size is split up to k-times larger distance.	**Min / Max** 0.1 - 2 **Default** 1
Water Reflections (boolean) Enables / disables screen space reflections on water.	**Default** false

Parameter (type)	Values
Preload Distance (float32) Terrain preloading distance around camera. **NOTE** `preload_dist` will be set to 0 for FOV < 0.49. Preloading is currently limited to load one more detail level than currently visible.	**Min / Max** 0 - 1000000 **Default** 0
Preload FOV Multiplier (float32) Terrain preloading around camera's frustum. Value of 1 disables preloading frustum. Standard scene subdivision still applies, so setting extreme values will not preload into infinity.	**Min / Max** 1 - 3 **Default** 1.5

6.4.17.12 Video / Graphics Settings / Viewport Settings / Vegetation Detail

Parameter (type)	Values
3D Trees and Bushes (boolean) Enables / disables 3D trees and bushes.	**Default** true

6.4.17.13 Video / Graphics Settings / Viewport Settings / Vegetation Detail / Tree Detail

Parameter (type)	Values
Tree Texture Resolution Limit (enum) Limits the best loaded texture resolution for biome trees. Total surface area is considered, means texture 2048x512 is equivalent to 1024x1024. Recommended value is Unlimited for graphics cards with >= 4 GB VRAM, or 2048 in more restricted memory conditions.	• 1/4k (256x256) • 1/2k (512x512) • 1k (1024x1024) • 2k (2048x2048) • Unlimited **Default** Unlimited

Parameter (type)	Values
Streamed Tree Draw Distance (float32) Draw distance of streamed tree objects. These are objects that are embedded in the terrain database. Value is distance in meters at which object of 1 meter radius would disappear when camera's field of view is 90 degrees.	**Min / Max** 100 - 2000 **Default** 170
Tree Draw Distance (float32) Draw distance of tree objects. These are objects such as runtime created trees. Value is distance in meters at which object of 1 meter radius would disappear when camera's field of view is 90 degrees.	**Min / Max** 100 - 2000 **Default** 170
Tree Detail (float32) The higher the value, the better the LODs of biome and placed trees used when close to the camera. Increasing the value will result in higher quality objects at further distances.	**Min / Max** 50 - 2000 **Default** 200
Biome Tree Fidelity (float32) Fidelity of trees. Value affects density of the vegetation, where lower value means lower density with the distance.	**Min / Max** 0.01 - 1 **Default** .5
Biome Tree Draw Distance (float32) Draw distance of trees. Value is distance in meters at which trees would disappear when camera's field of view is 90 degrees.	**Min / Max** 500 - 20000 **Default** 4200

6.4.17.14 Video / Graphics Settings / Viewport Settings / Vegetation Detail / Bush Detail

Parameter (type)	Values
Bush Texture Resolution Limit (enum) Limits the best loaded texture resolution for biome bushes. Total surface area is considered, means texture 2048x512 is equivalent to 1024x1024. Recommended value is Unlimited for graphics cards with >= 4 GB VRAM, or 2048 in more restricted memory conditions.	• 1/4k (256x256) • 1/2k (512x512) • 1k (1024x1024) • 2k (2048x2048) • Unlimited **Default** 1k (1024x1024)
Bush Detail (float32) The higher the value, the better the LODs of biome bushes used when close to the camera. Increasing the value will result in higher quality objects at further distances.	**Min / Max** 50.0000 - 2000.0000 **Default** 200.0000
Biome Bush Fidelity (float32) Fidelity of bushes. Value affects density of the vegetation, where lower value means lower density with the distance.	**Min / Max** 0.01000 - 1.0000 **Default** .50000
Biome Bush Draw Distance (float32) Draw distance of bushes. Value is distance in meters at which bushes would disappear when camera's field of view is 90 degrees.	**Min / Max** 100.0000 - 5000.0000 **Default** 1000.0000

6.4.17.15 Video / Graphics Settings / Viewport Settings / Vegetation Detail / Grass Detail

Parameter (type)	Values
Grass Texture Resolution Limit (enum) Limits the best loaded texture resolution for biome grass. Total surface area is considered, means texture 2048x512 is equivalent to 1024x1024. Recommended value is Unlimited for graphics cards with >= 4 GB VRAM, or 2048 in more restricted memory conditions.	• 1/4k (256x256) • 1/2k (512x512) • 1k (1024x1024) • 2k (2048x2048) • Unlimited **Default** 1k (1024x1024)
Grass Detail (float32) The higher the value, the better the LODs of biome grass used when close to the camera. Increasing the value will result in higher quality objects at further distances.	**Min / Max** 10 - 500 **Default** 100.0000
Biome Grass Fidelity (float32) Fidelity of grass. Value affects density of the vegetation, where lower value means lower density with the distance.	**Min / Max** 0.01 - 1 **Default** .5000
Biome Grass Draw Distance (float32) Draw distance of grass. Value is distance in meters at which grass would disappear when camera's field of view is 90 degrees.	**Min / Max** 10.0000 - 500.0000 **Default** 210.000-

6.4.17.16 Video / Graphics Settings / Viewport Settings / Object Detail

Parameter (type)	Values
Exterior Object Texture Resolution Limit (enum) Limits the best loaded texture resolution for exterior objects. Total surface area is considered, means texture 2048x512 is equivalent to 1024x1024. Recommended value is Unlimited for graphics cards with >= 4 GB VRAM, or 2048 in more restricted memory conditions.	• 1/4k (256x256) • 1/2k (512x512) • 1k (1024x1024) • 2k (2048x2048) • Unlimited **Default** Unlimited
Interior Object Texture Resolution Limit (enum) Limits the best loaded texture resolution for interior objects. Total surface area is considered, means texture 2048x512 is equivalent to 1024x1024. Recommended value is Unlimited for graphics cards with >= 4 GB VRAM, or 2048 in more restricted memory conditions.	• 1/4k (256x256) • 1/2k (512x512) • 1k (1024x1024) • 2k (2048x2048) • Unlimited **Default** Unlimited
Transparency Detail (float32) Distance where engine stops handling objects with transparency accurately and start to use approximate methods. The smaller the value, the better the performance, but the more likely we will see alpha order issues. Value is distance in meters at which object of 1 meter radius would switch the behavior when camera's field of view is 90 degrees.	**Min / Max** 0.1 - 200 **Default** 10
Streamed Static Draw Distance (float32) Draw distance of streamed static objects. These are objects that are embedded in the terrain database. Value is distance in meters at which object of 1 meter radius would disappear when camera's field of view is 90 degrees.	**Min / Max** 50 - 2000 **Default** 285
Static Draw Distance (float32) Draw distance of static objects. These are objects such as runtime created buildings. Value is distance in meters at which object of 1 meter radius would disappear when camera's field of view is 90 degrees.	**Min / Max** 100 - 2000 **Default** 285

Parameter (type)	Values
Land Draw Distance (float32) Draw distance of land objects. These are objects such as runtime created ground vehicles and lifeforms. Value is distance in meters at which object of 1 meter radius would disappear when camera's field of view is 90 degrees.	**Min / Max** 100 - 2000 **Default** 285
Water Draw Distance (float32) Draw distance of water objects. These are objects such as runtime created water vehicles. Value is distance in meters at which object of 1 meter radius would disappear when camera's field of view is 90 degrees..	**Min / Max** 100 - 2000 **Default** 550
Air Draw Distance (float32) Draw distance of air objects. These are objects such as runtime created aircraft and helicopters. Value is distance in meters at which object of 1 meter radius would disappear when camera's field of view is 90 degrees.	**Min / Max** 100 - 2000 **Default** 550
Streamed Wind Emitter Draw Distance (float32) Draw distance of streamed wind emitter objects. These are wind emitters that are embedded in the terrain database. Value is distance in meters at which object of 1 meter radius would disappear when camera's field of view is 90 degrees.	**Min / Max** 1 - 200 **Default** 5
Wind Emitter Draw Distance (float32) Draw distance of wind emitter objects. These are objects such as runtime created wind emitters. Value is distance in meters at which object of 1 meter radius would disappear when camera's field of view is 90 degrees.	**Min / Max** 0 - 200 **Default** 5
Streamed Force Emitter Draw Distance (float32) Draw distance of streamed force emitter objects. These are force emitters that are embedded in the terrain database. Value is distance in meters at which object of 1 meter radius would disappear when camera's field of view is 90 degrees.	**Min / Max:** 0.000000 - 200.000000 **Default:** 40.000000

Parameter (type)	Values
Force Emitter Draw Distance **(float32)** Draw distance of force emitter objects. These are objects such as runtime created wind emitters. Value is distance in meters at which object of 1 meter radius would disappear when camera's field of view is 90 degrees.	**Min / Max:** 0.000000 - 200.000000 **Default:** 40.000000
Streamed Damage Area Draw Distance **(float32)** Draw distance of streamed damage areas objects. These are damage areas that are embedded in the terrain database. Value is distance in meters at which object of 1 meter radius would disappear when camera's field of view is 90 degrees.	**Min / Max:** 0.000000 - 2000.000000 **Default:** 500.000000
Damage Area Draw Distance **(float32)** Draw distance of dedicated damage area objects. These are objects such as runtime created damage areas. Value is distance in meters at which object of 1 meter radius would disappear when camera's field of view is 90 degrees.	**Min / Max:** 0.000000 - 2000.000000 **Default:** 500.000000
Streamed Static Fidelity **(int32)** The higher the value, the better the LODs of streamed static objects when close to the camera. These are objects that are embedded in the terrain database. Increasing the value will result in higher quality objects at further distances.	**Min / Max** 10 - 3600 **Default** 300
Static Fidelity **(int32)** The higher the value, the better the LODs of static objects when close to the camera. These are objects such as runtime created buildings. Increasing the value will result in higher quality objects at further distances.	**Min / Max** 10 - 3600 **Default** 300
Land Fidelity **(int32)** The higher the value, the better the LODs of land objects when close to the camera. These are objects such as runtime created ground vehicles and lifeforms. Increasing the value will result in higher quality objects at further distances.	**Min / Max** 10 - 3600 **Default** 300

Parameter (type)	Values
Water Fidelity **(int32)** The higher the value, the better the LODs of water objects when close to the camera. These are objects such as runtime created water vehicles. Increasing the value will result in higher quality objects at further distances.	**Min / Max** 10 - 3600 **Default** 300
Air Fidelity **(int32)** The higher the value, the better the LODs of air objects when close to the camera. These are objects such as runtime created aircraft and helicopters. Increasing the value will result in higher quality objects at further distances.	**Min / Max** 10 - 3600 **Default** 300
Air Dot Size **(float32)** Pixel size of a dot representing an object that is too small to be visible with current resolution.	**Min / Max** 0 - 4 **Default** 1
Point Cloud Detail **(float32)** Draw distance of point cloud data. Value is distance in meters up to which a segment with one meter size is split into smaller segments when camera's field of view is 90 degrees. Segment with k-meter size is split up to k-times larger distance.	**Min / Max** 0.1 - 2 **Default** 0.262
Cloud Detail **(float32)** Draw distance of clouds. Value is distance in meters up to which a segment with one meter size is split into smaller segments when camera's field of view is 90 degrees. Segment with k-meter size is split up to k-times larger distance.	**Min / Max** 0.5 - 4.0000 **Default** 1.047
High Detail Volumetric Clouds **(boolean)** Enable / disable high detail shading of volumetric cloud layers.	**Default** true
Building Detail **(float32)** Draw distance of geometry layer (buildings per segment). Value is distance in meters up to which a segment with one meter size is split into smaller segments when camera's field of view is 90 degrees. Segment with k-meter size is split up to k-times larger distance.	**Min / Max** 0.0025 - 4 **Default** .25

6.4.17.17 Video / Graphics Settings / Viewport Settings / Particle Detail

Parameter (type)	Values
Particle Effect Fidelity **(float32)** The lower the value the better LODs of particles are selected closer to the camera.	**Min / Max** 0 - 6 **Default** .0007
Particle Effect Detail **(float32)** The higher the value the better lighting quality of particles.	**Min / Max:** 0.000000 - 1.000000 **Default:** 0.500000
Particle Draw Distance **(int32)** The lower the value (slider moved to the right) the higher the draw distance and detail of the particles. This value is a coefficient that is also affected by field of view.	**Min / Max** 1 - 100 **Default** 10

6.4.17.18 Video / Graphics Settings / Viewport Settings / Light Detail

Parameter (type)	Values
Streamed Light Draw Distance **(float32)** Draw distance of streamed light objects. These are lights that are embedded in the terrain database and it can be understood as lightmap transition distance. Value is distance in meters at which object of 1 meter radius would disappear when camera's field of view is 90 degrees.	**Min / Max** 1 - 40 **Default** 20
Light Draw Distance **(float32)** Draw distance of light objects. These are objects such as runtime created lights. Value is distance in meters at which object of 1 meter radius would disappear when camera's field of view is 90 degrees.	**Min / Max** 1.000 - 2000.000 **Default** 170.0000
Streamed Emissive Plane Draw Distance **(float32)** Draw distance of streamed emissive planes of light reflectors. Value is distance in meters at which object of 1 meter radius would disappear when camera's field of view is 90 degrees.	**Min / Max** 1 - 2000.000 **Default** 170.000

Parameter (type)	Values
Emissive Plane Draw Distance (float32) Draw distance of dynamic emissive planes of light reflectors. Value is distance in meters at which object of 1 meter radius would disappear when camera's field of view is 90 degrees.	**Min / Max** 1.000 - 2000.000 **Default** 170.000

6.4.17.19 Video / Graphics Settings / Viewport Settings / Post-Process Effects

Parameter (type)	Values
Ambient Occlusion (enum) Method used for ambient occlusion. This improves the visual quality of lighting, and shadows on objects.	• Disabled • SSAO • HBAO **Default:** SSAO
Motion Blur (boolean) Enable / disable motion blur postprocess effect.	**Default:** false

6.4.17.20 Video / Graphics Settings / Viewport Settings / Draw Features

Parameter (type)	Values
Sky (boolean) Controls whether the skydome and atmosphere are enabled for rendering.	**Default:** true
Sun (boolean) Controls whether the Sun is enabled for rendering.	**Default:** true
Moon (boolean) Controls whether the Moon is enabled for rendering.	**Default:** true
Stars (boolean) Controls whether the stars are enabled for rendering.	**Default:** true

Parameter (type)	Values
Ground **(boolean)** Controls whether the ground is enabled for rendering.	**Default:** true
Water **(boolean)** Controls whether the water is enabled for rendering.	**Default:** true
Biome Trees **(boolean)** Controls whether biome trees are enabled for rendering.	**Default:** true
Biome Bushes **(boolean)** Controls whether biome bushes are enabled for rendering.	**Default:** true
Biome Grass **(boolean)** Controls whether biome grass is enabled for rendering.	**Default:** true
Geometry **(boolean)** Controls whether procedural geometry that is included in the terrain, such as buildings, is enabled for rendering.	**Default:** true
Point Clouds **(boolean)** Controls whether point cloud objects are enabled for rendering.	**Default:** true
Objects **(boolean)** Controls whether streamed and dynamic objects, such as lifeform and platforms, are enabled for rendering.	**Default:** true
Lights **(boolean)** Controls whether the sun is enabled for rendering.	**Default:** true
Particles **(boolean)** Controls whether lights are enabled for rendering.	**Default:** true

Parameter (type)	Values
Clouds **(boolean)** Controls whether clouds are enabled for rendering.	**Default:** true
Precipitation **(boolean)** Controls whether precipitation effects are enabled for rendering.	**Default:** true

6.4.17.21 Video / Graphics Settings / Compositor Settings / Post-Process

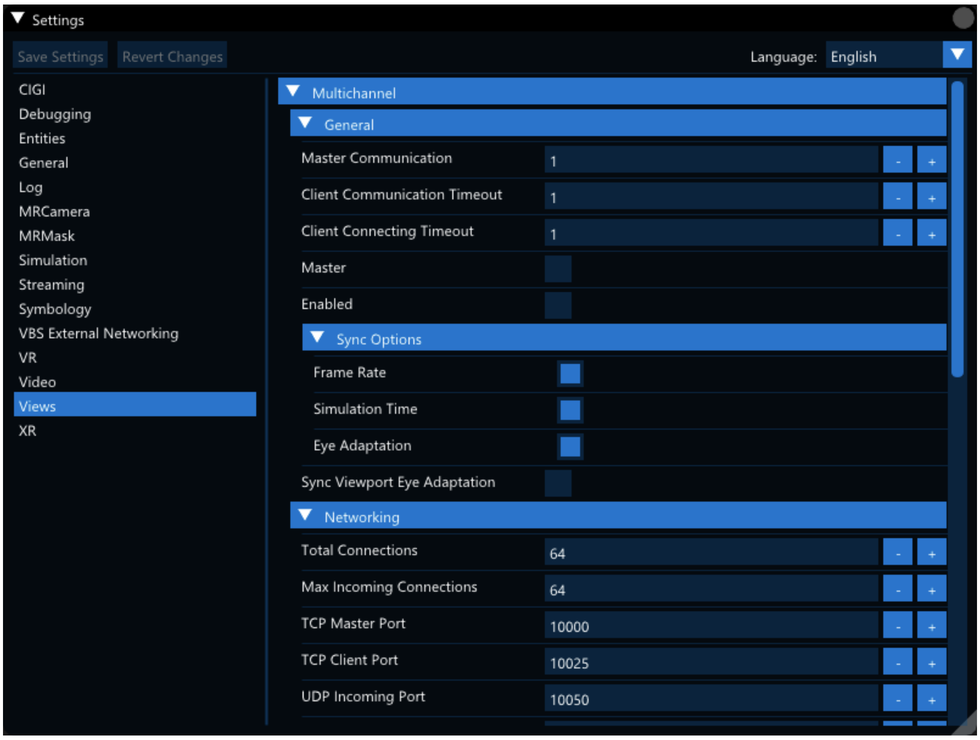
Parameter (type)	Values
Bloom Spread Level **(int32)** For the Post Process Bloom Flare Type, controls the maximum amount of spread of the effect.	**Min / Max** 2 - 10 **Default** 10
Bloom Strength **(float32)** Strength of blooming effect when used by compositor. Required for local tonemapping.	**Min / Max** 0 -2 **Default** 1
Lens Effects Strength **(float32)** Camera lens effects like dirt and flare. Affected by bloom spread level.	**Min / Max** 0 - 2 **Default** 1

6.4.18 Views Settings

To synchronize multiple view channels, designate one of the VBS Blue IG Clients as the master client. IG Clients wait for a message from the master client before continuing processing. These clients communicate with the master through TCP and UDP network messages, while the master client communicates to all of its clients through multi-cast network messages.

Set up multi-channel views using either of the following methods:

- **Settings UI** - Press **Tab** to open the VBS Blue IG Settings (on page 201) and make adjustments, as required.



- **Edit XML file** - Open the `Views.xml` file, located in the following directory and edit, as required:

  `%LOCALAPPDATA%\Bohemia Interactive Simulations\`
  `VBS Blue IG\`*`version`*`\Settings\`

To setup multi-channel synchronization, **follow these steps**:

> **NOTE**
> There should be only one master client in any group of VBS Blue IG clients.

1. Enable the multichannel functionality for each IG instance. This should be done regardless of whether the instance is the master or client, and can be enabled using either of the following options:
   - Select **Views / Multichannel / General**, and click the **Enabled** checkbox.
   - Set the `<Enabled>` flag to `true` in the `Views.xml` file.
2. Designate a master client using one of the following options:
   - Select **Views / Multichannel / General**, and click the **Master** checkbox.
   - Launch the desired client with the `-master` command-line parameter.
   - On the target client computer, set the `<Master>` setting flag to `true` in the `Views.xml` file.
3. Determine and set a Multicast IP Address and port. All clients including the master client should have the same multicast address and port set.
4. Set the TCP address of the client instances to match the IP address of the PC where the master is running.

   > **NOTE**
   > In the field **Networking / Master Address**, the default value of the master instance is 127.0.0.1 (localhost). Although this default value can remain in the field, the actual IP address of the master client PC must be used in the client instances.

   All other ports only need to be adjusted, if they are already being used by another application.
5. Click **Save Settings**.
6. Restart VBS Blue IG.

Keep the following considerations in mind when using multi-channel synchronization:

- The master client and clients can be started and loaded independently.
- While VBS Blue IG loads, some clients may attempt to connect to the master and experience some time-outs. This stabilizes as soon as all clients finish loading.
- For optimal performance, synchronization works best in a low network latency environment, with VBS Blue IG running in full-screen mode and VSync turned on. In addition, the **fixed_screen_space_size_calc** option should be selected for all clients in the Video Settings (on page 340) menu.
- Master clients are not automatically found and determined, if no master can be found by a client, no deliberate synchronization occurs.

The tables below describe the type and purpose for each setting.

6.4.18.1 Views / Multichannel / General

Parameter (type)	Values
Master Communication **(uint8)** Max amount of time (seconds) a client waits for master render messages before it times out. Range 1 - 60	**Min / Max** 1 - 60 **Default** 1
Client Communication Timeout **(uint8)** Max amount of time (seconds) a master waits for client ready messages before it times out. Range 1 - 60	**Min / Max** 1 - 60 **Default** 1
Client Connecting Timeout **(uint16)** Max amount of time (seconds) a master waits for client first ready message before it times out. Range 1 - 300	**Min / Max** 1 - 300 **Default** 1
Master **(boolean)** Enables / Disables Master client mode	**Default** false
Enabled **(boolean)** Enables / Disables Multi-Channel functionality for this IG.	**Default** false

6.4.18.2 Views / Multichannel / General / Sync Options

Parameter (type)	Values
Frame Rate **(boolean)** When enabled, software based synchronization delays rendering of each frame until all members of the multichannel group are ready to render it. This effectively limits the frame rate to the lowest common denominator.	**Default** true
Simulation Time **(boolean)** When enabled, the current Simulation Time from the Master will be applied to the multichannel group.	**Default** true
Eye Adaptation **(boolean)** When enabled, Eye Adaptation values from the Master will be applied to the multichannel group.	**Default** true
Sync Viewport Eye Adaptation **(boolean)** If true, synchronizes eye adaptation between viewports on this VBS Blue IG instance without the overhead of IGMultiChannel networking.	**Default** false

6.4.18.3 Views / Multichannel / Networking

Parameter (type)	Values
Total Connections (uint16)	**Default** 64
Max Incoming Connections (uint16)	**Default** 64
TCP Master Port (uint16)	**Default** 10000
TCP Client Port (uint16)	**Default** 10025
UDP Incoming Port (uint16)	**Default** 10050
UDP Outgoing Port (uint16)	**Default** 10051
Multicast Port (uint16)	**Default** 10052
TCP Port Range (uint16)	**Default** 24
Multicast Address (string)	**Default** 225.0.0.100
Multicast NIC (string)	**Default** 0.0.0.0
Master Address (string)	**Default** 127.0.0.1

6.4.18.4 Views / Effects

Parameter (type)	Values
Apply Velocity Effect (boolean) Enables / Disables Velocity Effect on Camera	**Default** true

6.4.18.5 Views / Warping / Scalable

Parameter (type)	Values
Position Scale **(float64)** The multiplier to convert from the units defined in the configuration file, to meters. For example, 1.0 = meters, 0.001 = millimeters.	**Default** 0.01000
Warp Files **(string)** Defines the path to the warp file, ordered by search order. The warp data will attempt to load from the first path, and if not found, continue to the next, until all WarpFile paths are exhausted. Paths can either be absolute or relative to WarpComponent directory	**Default** `C:\Program Files\Scalable Display \DEI\LocalCalibration\ ScalableData.ol`

6.4.18.6 Views / Warping / Dome Projection

Parameter (type)	Values

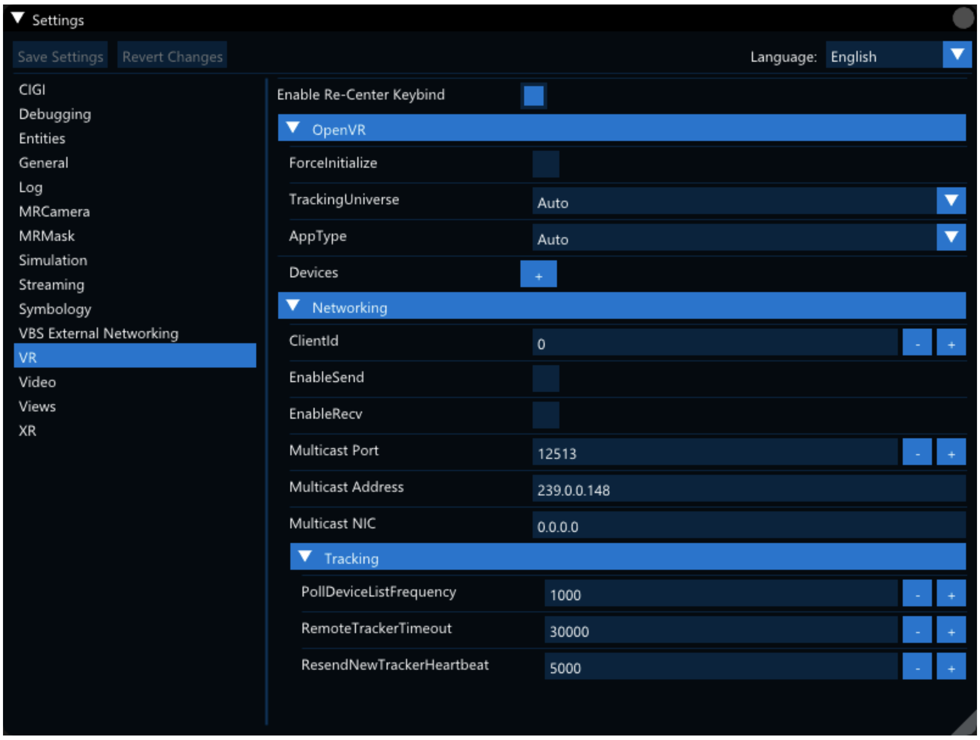
Warp Files
(string)
Defines the path to the warp file, ordered by search order.
The warp data will attempt to load from the first path, and if not found, continue to the next, until all WarpFile paths are exhausted.
Paths can either be absolute or relative to WarpComponent directory
	Default
	`C:\DomeProjection\data\config.xml`
	`DomeProjection\config.xml`

Apply Frustum **(boolean)** If True, the view will have its frustum overridden by the frustum defined by the warp files.Default	**Default** true
Apply Position **(boolean)** If True, the view will have its position offset by the position defined by the warp files.	**Default** true
Apply Orientation **(boolean)** If True, the view will have its orientation offset by the orientation defined by the warp files.	**Default** true
PositionScale **(float64)** The multiplier to convert from the units defined in the configuration file, to meters. For example, 1.0 = meters, 0.001 = millimeters.	**Default** .001000

6.4.19 VR Devices Settings

Customize the settings for VR Devices for a simulation host using either of the following methods:

- **Settings UI** - Press **Tab** to open the VBS Blue IG Settings (on page 201) and make adjustments, as required.



- **Edit XML file** - Open the `VR.xml` file, located in the following directory and edit, as required:

  `%LOCALAPPDATA%\Bohemia Interactive Simulations\`
  `VBS Blue IG\`*`version`*`\Settings\`

The tables below describe the type and purpose for each setting.

6.4.19.1 VR

Parameter (type)	Values
Enable Re-Center Keybind (boolean) Enables the keybind used to re-center a connected head mounted device. Key Combination: **LCTRL** + **LShift**.	**Default** true (enabled)

6.4.19.2 VR / OpenVR

OpenVR/SteamVR initialization settings.

Parameter (type)	Values
ForceInitialize **(boolean)** If true, this forces the component to initialize OpenVR interface as well as shutting it down. This is only used if either a non-OpenVR HMD or no HMD is in use. **NOTE** Changing may require an application restart.	**Default** false
TrackingUniverse **(enum)** Style of tracking origin for the reported poses. Only applies if OpenVR is forced enabled.	• Auto • Seated • Standing • RawAndUncalibrated **Default** Auto
AppType **(enum)** OpenVR application type for forced OpenVR init; ignored if OpenVR HMD is active.	• Auto • Background • Other **Default** Auto

6.4.19.3 VR / Networking

Shared / Networked tracker settings. Corresponds to a new feature called TrackerNet.

Parameter (type)	Values
ClientId **(uint16)** Unique client ID for this local instance	**Default** 0
EnableSend **(boolean)** Enables sending local tracker data to network.	**Default** false
EnableRecv **(boolean)** Enables receiving remote tracker data from network.	**Default** false
Multicast Port **(uint16)** Multicast port number to send / receive tracking data.	**Default** 12513
Multicast Address **(string)** Multicast address to send tracking data, as well as listen and receive remote data.	**Default** 239.0.0.148
Multicast NIC **(string)** Multicast NIC to send and receive multicast tracking data.	**Default** 0.0.0.0

6.4.19.4 VR / Networking / Tracking

Allow clients to share VR devices.

Parameter (type)	Values
PollDeviceListFrequency (uint32) Maximum time (in milliseconds) before sending out all known tracker positions, regardless if the tracker has been turned off or has been inactive.	**Min / Max** 0 - 1000000 **Default** 1000
RemoteTrackerTimeout (uint32) Time (in milliseconds) of inactivity before local system automatically removes remote tracked devices. Assumes remote client has gone offline before it could send remove notification.	**Min / Max** 0 - 1000000 **Default** 30000
ResendNewTrackerHeartbeat (uint32) Time (in milliseconds) before local system sends out verbose tracker information (besides just transforms), in case remote client has entered recently.	**Min / Max** 0 - 1000000 **Default** 5000

6.4.20 XR Settings

Customize the XR Training Platform settings using either of the following methods:

- **Settings UI** - Press **Tab** to open the VBS Blue IG Settings (on page 201) and make adjustments, as required.

- **Edit XML file** - Open the `XR.xml` file located in the following directory and edit, as required:

  `%LOCALAPPDATA%\Bohemia Interactive Simulations\`
  `VBS Blue IG\`*`version`*`\Settings\`

The tables below describe the type and purpose for each setting.

6.4.20.1 XR

XR Training Platform settings.

Parameter (type)	Values
Enabled **(boolean)** Toggles the functionality of this component.	**Default** false

When toggled on, additional functionality becomes available in the Debug UI > XR Training Platform option. For more information and example usage of this function, see Example Weapon Configuration - M4 (on page 126).

6.4.20.2 XR / TrackingSpace / Transform / Translation

Parameter (type)	Values
x **(float64)**	**Default** 0
y **(float64)**	**Default** 0
z **(float64)**	**Default** 0

6.4.20.3 XR / TrackingSpace / Transform / Rotation

Euler angles (yaw, pitch, roll) in degrees.

Parameter (type)	Values
yaw **(float64)**	**Min / Max** -180.0000 - 180.0000 **Default** 0
pitch **(float64)**	**Min / Max** -180.0000 - 180.0000 **Default** 0
roll **(float64)**	**Min / Max** -180.0000 - 180.0000 **Default** 0

6.4.20.4 XR / Lasers

Global XR Laser settings.

Parameter (type)	Values
Mode **(enum)** Laser behavior.	• Default • Optimized Intersection • No Intersection **Default** Optimized Intersection
IntersectionMinDetail **(uint8)** Laser ray intersection minimum detail; only valid if LaserMode is set to "Optimized Intersection".	**Min / Max** 0 - 23 **Default** 8
IntersectionMaxDetail **(uint8)** Laser ray intersection maximum detail; only valid if LaserMode is set to "Optimized Intersection".	**Min / Max** 0 - 23 **Default** 23
MaxDistance **(float32)** Max Distance (in meters) for laser beam, [1,1000000]. Larger values may impact performance if intersections are enabled.	**Min / Max** 0.0000 - 1000000.0000 **Default** 5000.0000
IntersectFwdStart **(float32)** Z+ (forward) offset for intersection tests. Increase this to prevent the laser from intersecting close models.	**Min / Max** 0.0000 - 1000000.0000 **Default** 5000.0000
LightIntensity **(float32)** Intensity of light source at intersection point; uses laser color. Set to 0 to disable light	**Min / Max** 0.0000 - 1000000.0000 **Default** 5000.0000
LightIntensity **(float32)** Intensity of light source at intersection point; uses laser color. Set to 0 to disable light	**Default** true

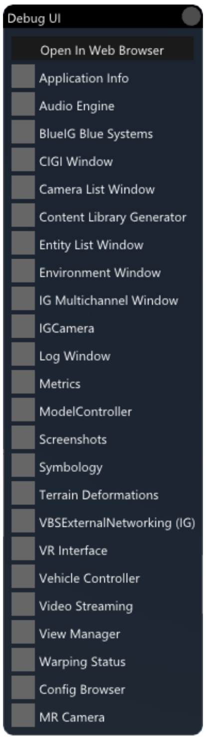
Parameter (type)	Values
Flare **(boolean)** Toggles the visual light flare at point of laser intersection.	**Default** true
IgnoreOwnship **(boolean)** Toggles whether or not the laser intersection ignores the ownship model. May be preferred if the laser passes through open doorways or near animated geometry.	**Default** true
IgnoreParticles **(boolean)** Toggles whether or not the laser intersection ignores particles. May be preferred under heavy rotorwash, smoke or other particle effects.	**Default** true

6.4.20.5 XR / Platform

Parameter (type)	Values
AutoAttachView **(boolean)** Auto attaches view to ownship platform.	**Default** false
SimulateTracers **(boolean)** Simulates tracers fired from this platform.	**Default** false
PlatformMuzzleScaling **(boolean)** Uses scaled parent offsets when sending muzzle transformations.	**Default** false

6.5 Debug UI

Use the Debug UI to access tools for runtime testing and debugging VBS Blue IG scenarios. Debug UI can either be accessed within VBS Blue IG, or remotely via a web browser. Accessing remotely via a web browser can be useful with debugging and configuring multiple PCs that may not necessarily have input available.



Follow these steps:

1. Access the Debug UI:
    - Press backquote ` / tilde ~ on the keyboard to open the Debug UI.
    - **Optional**: Access the Debug UI remotely. For more information, see Access Debug UI remotely (on the next page).
2. Select from the control and configuration options. For more information, see Configuration Options (on the next page).

> **NOTE**
>
> Unlike the VBS Blue IG Settings (on page 201) menu, settings or adjustments made in Debug UI are not saved when VBS Blue IG is closed.

> **NOTE**
>
> More advanced Debug UI options are available by enabling them within the VBS Blue IG Settings UI. For more information, see VBS Blue IG Settings (on page 201).

6.5.1 Access Debug UI remotely

1. In the Debug UI window, select the **Open In Web Browser** button or navigate to `http://<ip>:7890` in a web browser.

   Once opened in a web browser, both Debug UI and the Settings UI are available to remotely access and control.

   To configure a different port from the default port, modify `RemotePort` in:

   `\IG_Installation\Components\DebugUIDX11\Settings\Settings.xml`.

   > **NOTE**
   > Setting the port to 0 disables this feature.

   > **NOTE**
   > If preferred, you may also disable the Debug UI and Settings UI so that they are only available remotely in the web browser by toggling **Render in Application**.

2. **Copy to Clipboard** - Some Debug UI windows offer the option **Copy to Clipboard** for certain data. This can be also used via the Debug UI session opened in a browser. Any information copied this way can be accessed by the **Show Clipboard Data** and then pasted into a file on the remote machine.

6.5.2 Configuration Options

- **Application Info** - Provides detailed information about the installed version of VBS Blue IG. Use the **Copy to Clipboard** button to quickly paste information about your build into a support ticket.
- Audio Engine (on page 379) - Provides basic support for playing and adding sound effects.
- **Blue IG Blue Systems** - The Force Fallback Camera toggle is used to override views, and allows users to move the camera freely with the default camera controls. In **View Manager > Fallback View**, the **Copy View XML** button copies the XML to the clipboard.
- CIGI Window (on page 386) - The CIGI window displays a list of all the CIGI mapping files being used and provides a reload option to change mappings at runtime without restarting VBS Blue IG.
- Camera List Window (on page 383) - Allows switching between views, setting current date and time for the scene, coordinates for a camera jump.
- Content Library Generator (on page 388) - Content Library Generator window provides the ability to export content library XML files and screenshots for all content loaded in VBS Blue IG.

- Entity List Window (on page 389) - Displays information about entities.
- Environment Window (on page 393) - Controls the weather settings.
- IG Multichannel Window (on page 398) - Displays the VBS Blue IG client / master information, depending on the VBS Blue IG setup.
- **IG Camera** - The **Allow Camera Under Surface** toggle is used to allow the fallback camera under the terrain surface.
- Log Window (on page 402) - The log window outputs status information about VBS Blue IG.
- Metrics (on page 404) - Displays real time views of FPS data and other product-related metrics.
- Model Controller (on page 406) - The Model Controller window allows you to create, delete and modify placed models.
- Screenshots (on page 408) - Provides quick capture of screen images of the current IG instance.
- Symbology (on page 409) - Displays debugging information about CIGI-based Symbology.
- **Terrain Deformations** - Provides the option to reset volumetric snow deformations generated in VBS Blue IG by a VBS or CIGI host. Use the **Reset Soft Surface Deformations** button to revert a snow-covered terrain to its original state.
- VBS External Networking (on page 411) - The VBS Interop client networking window, providing status information about the networking session.
- VR Interface (on page 425) - Provides information, diagnostic tools, and positional re-centering for head-mounted displays (HMDs) and VR-related devices supported by VBS Blue IG.
- **Vehicle Controller** - Provides information regarding selected vehicles in the **Entity List** Window. Specifically, available vehicle lights can be viewed and configured for the selected entity.
- Video Streaming (on page 415) - The Video Streaming Debug UI window allows you to enable streams of a scenario. Users can specify which render target or view to be streamed using the Debug window. Additionally, output can be customized further using the Settings UI Streaming Settings (on page 314).
- View Manager (on page 416) - Allows you to view and configure all the currently registered views in the current instance of VBS Blue IG.
- Warping Status (on page 427) - Controls for both Generic and Scalable Warping solutions (see Generic Warping Setup (on page 133) and Scalable Warping Setup (on page 140)).
- Config Browser (on page 431) - Provides views of the configuration of any model.
- MR Camera (on page 432) - Allows user to integrate sourced imagery from cameras, videos, or images and inject them into a rendered Mixed Reality / Virtual Reality scene.

6.5.3 Audio Engine

VBS Blue IG contains basic support for playback of sound effects. The application allows for playing back and adding simple stereo sounds, or more advanced positional 3D audio with up to 7.1 channel support.

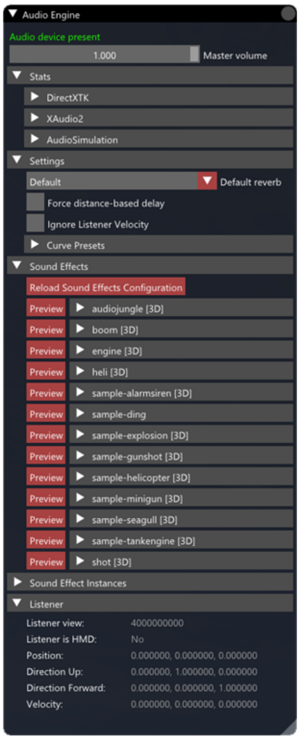
Use these procedures for making use of the Audio Engine:

- Debug UI Testing (below)
- Adding Sounds (on the next page)
- CIGI Playback (on page 381)

6.5.3.1 Debug UI Testing

The Audio Engine window within the Debug UI provides the following options:

- Monitoring of audio usage stats.
- Modify and test runtime audio settings such as reverb, delay, and velocity.
- Preview sound effects.



Follow these steps:

1. Press backquote ` / tilde ~ to access the Debug UI (on page 376).
2. Select **Audio Engine**.

3. Expand any of the following:

Parameters Section	Description
Master Volume	Adjust sound level of master volume setting.
Stats	Select any of the following to view statistical data for the item: • **DirectXTK** • **XAudio2** • **AudioSimulation**
Settings	• **Default Reverb** - Select a Default Reverb option type from the drop-down list. • **Force distance-based delay** - Click to enable / disable. • **Ignore Listener Velocity** - Click to enable / disable.
Sound Effects	• **Reload Sound Effects Configuration** - Reloads the configuration of all sound effects. **NOTE** Pre-existing sound effects will not be effected by reloading, only new effect instances added. To add new files, see Adding Sounds (below). • **Preview** - Preview the sound effect by playing it as a one-shot sound.
Sound Effect Instances	Displays any instance of sound effects in use.
Listener	Displays information about Listeners added to an IG instance.

6.5.3.2 Adding Sounds

Follow these steps:

1. Add a .wav file containing your sound to the following directory:

   `\IG_Installation\Components\AudioEngine\Sounds\`

   If the Sounds directory does not exist, create it.

2. Add a `MySounds.xml` file (the name of the XML file does not matter) containing a group of mappings between the name of the sound and the .wav file, as well as additional properties for the sound effects. A basic sample containing the available parameters is listed below.

```
<?xml version="1.0"?>
<SoundEffects>
  <SoundEffect3D name="heli" file="heli.wav" looping="true" volume="1.0"
pitch="0.0"
   innerRadius="0.0" innerRadiusAngle="0.0" curveDistanceScaler="5000"
dopplerScaler="1.0" />
  <SoundEffect3D name="boom" file="boom.wav" curveDistanceScaler="5000.0" />
  <SoundEffect3D name="shot" file="shot.wav" curveDistanceScaler="1200.0" />
  <SoundEffect3D name="engine" file="engine.wav" curveDistanceScaler="1200.0"
looping="true" volume="1.0" pitch="0.0" pan="0.0" />
  <SoundEffect name="mouseclick" file="click.wav" />
</SoundEffects>
```

6.5.3.3 CIGI Playback

Sound effects can be controlled by a CIGI host. Sounds are created with the **Entity Control** packet, using a **Sound** entity type. The **Animation State** parameter (in the **Animation Control** packet in CIGI 4.0) controls playback.

Follow these steps:

1. Open the Cigi Protocol settings at:

   `%LOCALAPPDATA%\Bohemia Interactive Simulations\VBS Blue IG\<version>\Settings\CIGI.xml`.

2. Modify the Queries setting to `<Queries Enabled="false">`.

3. Add a new entity mapping in the file, using the sound identifier `<sound name>`.

```
cigi-entity-type:50000 > sound:heli
cigi-entity-type:50001 > sound:boom
cigi-entity-type:50002 > sound:shot
cigi-entity-type:50003 > sound:engine
cigi-entity-type:50004 > sound:mouseClick
```

4. Send an **Entity Control** packet with the mapping identifier as the Entity Type (for example, 50000), and the **Animation State** parameter = 2 (Play).

   The sound should play with the following conditions:

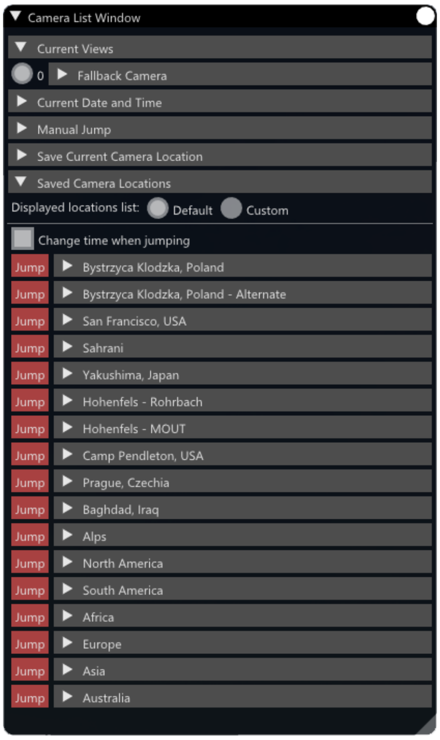
   - 3D sound effect - Plays in the sound entity's location.
   - Non-positional sound effect - Plays in all locations.

   When the sound stops, or a non-looping sound finishes playing, an **Animation Stop Notification Packet** is sent to the Host. The Host should then destroy the entity. This is a similar process to how particle effects function.

6.5.4 Camera List Window

The Camera List Window allows you to:

- Switch between views.
- Set the current date and time for the scene.
- Provide coordinates to which the camera can jump.
- Save the current camera location.
- Open saved camera locations.



Follow these steps:

1. Press backquote (`) / tilde (~) to access the Debug UI (on page 376).
2. Select **Camera List Window**.

3. Set the following parameters:

<table>
<tr><th>Parameters Section</th><th>Description</th></tr>
<tr><td>Current Views</td><td>Switch between views.
Click any of the available views to select it.
Fallback Camera - this option allows you to select an entity from the Entity List Window.
Expand the selected view to see the geodetic ellipsoid and geoid positions, and orientation. Click Copy to copy the respective values to the clipboard.</td></tr>
<tr><td>Current Date and Time</td><td>Set the current time and date.
Year (use + / -, or enter the year in the YYYY format)
Month (use the slider to set the month)
Day (use the slider to set the day)
Hour (use the slider to set the hour)
Minute (use the slider to set the minute)
Second (use the slider to set the second)</td></tr>
<tr><td>Manual Jump</td><td>Make the camera jump to the specified coordinates.
Coordinate - For latitude, longitude, and altitude, use + / -, or enter the coordinates manually.
Rotation - For yaw and pitch, use + / -, or enter the values manually.
Click Jump for the camera to jump to the specified coordinates.</td></tr>
<tr><td>Save Current Camera Location</td><td>Save the current camera location.
Location Name - Enter the location name, under which you want to save it.
Click Save Location to save it (the location then appears under Saved Camera Locations).
You may also create or delete camera locations by directly modifying the <code>CameraLocations.xml</code> file. For more information, see Camera Locations (on page 435).

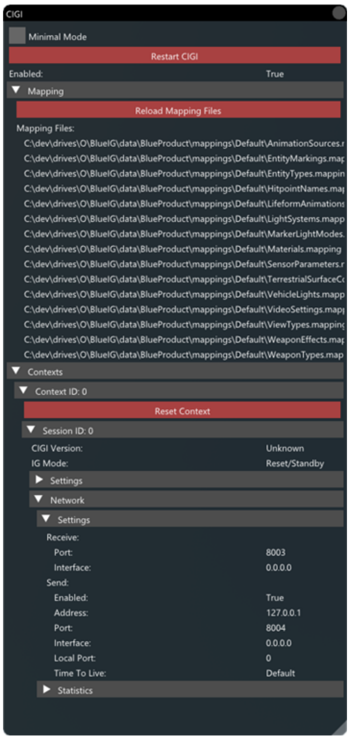
NOTE
The file <code>CameraLocations.xml</code> is only created after a custom location has been saved in the Camera List Window from the Debug UI.</td></tr>
</table>

Parameters Section	Description

Saved Camera Locations	Open saved camera locations.
	• **Change Time when Jumping** - Check this to use the date and time saved in the location.
	• **Default** and **Custom** changes between default locations distributed with the VBS Blue IG installation and custom locations saved locally to the current computer.
	• Scroll to the saved location:
	  ◦ Expand it to see the location coordinates, rotation values, and date and time.
	  ◦ Click **Jump** to jump to the location.
	  ◦ Click **[X]** and then **[Y]** to delete the location. Only custom locations can be deleted.

6.5.5 CIGI Window

The CIGI window allows you to restart and reload preferred versions of CIGI in the current instance of VBS Blue IG.



Follow these steps:

1. Press backquote (`) / tilde (~) to access the Debug UI (on page 376).
2. Select **CIGI Window**.

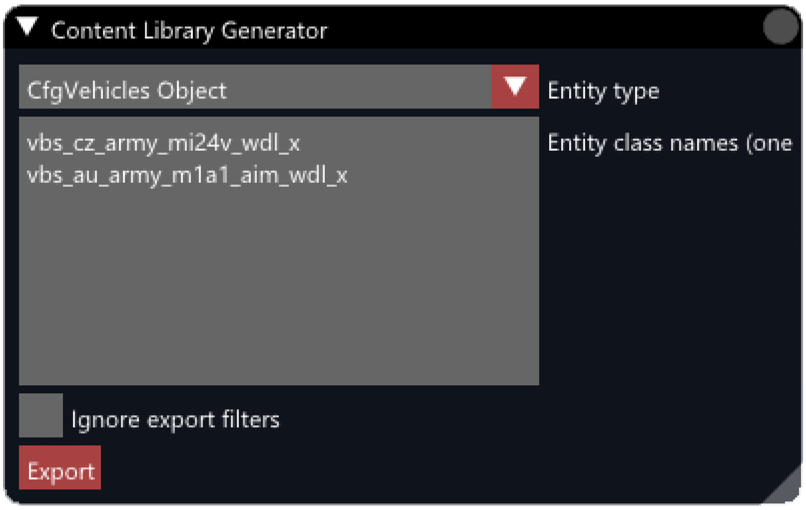
3. Choose from the following options:

<table>
<tr><th>Parameters Section</th><th>Description</th></tr>
<tr><td>Minimal Mode</td><td>Enable to create transparent overlay of UI onto scenario.</td></tr>
<tr><td>Restart CIGI</td><td>Reloads all settings and resets CIGI, which also deletes all CIGI entities.</td></tr>
<tr><td>Mapping</td><td>Click the dropdown to show the Mapping Files and the following option:
Reload Mapping Files - Reloads all mapping files and helps change mappings at runtime without restarting.</td></tr>
<tr><td>Contexts</td><td>Click the dropdown to show the Contexts and the following option:
Reset Content - Resets this scene context. This performs the same function as if a Host sent an IG Control packet o this context with IG Mode = Reset.
Session ID
Settings - Receive / Send settings.
Network
Settings - Network settings data for current session.
Statistics - Network data for current session statistics.</td></tr>
</table>

6.5.6 Content Library Generator

The Content Library Generator window provides the ability to export content library XML files and screenshots for all content loaded in VBS Blue IG. It is intended to be used for custom content since the default content provided in VBS Blue IG is already exported and available in:

`\IG_Installation\docs\ContentLibrary.zip`.



Follow these steps:

1. Press backquote ` / tilde ~ to access the Debug UI (on page 376).
2. Select **Content Library Generator**.
3. Choose from the following options:

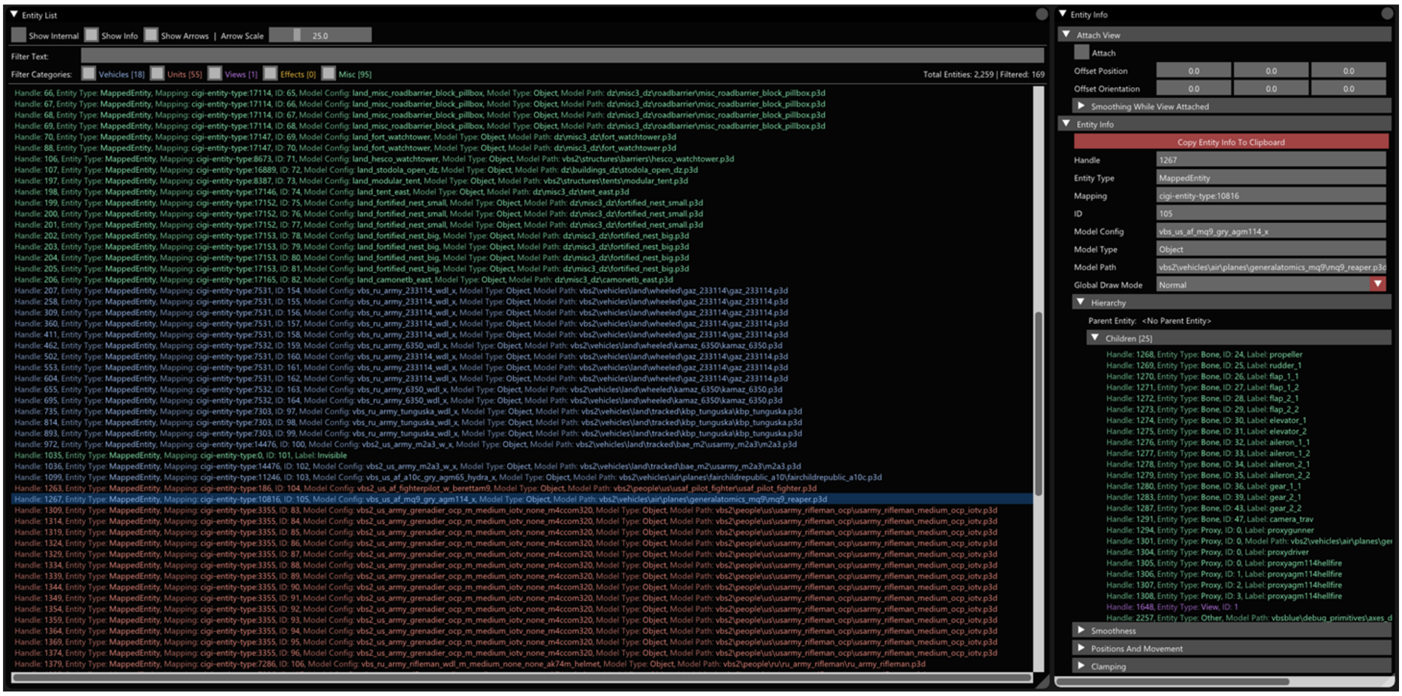
Parameters Section	Description
Entity Type	The super class of the objects to export.
Entity class names	List the class names of each object to export, with one class name per line.
Ignore export filters	Forces objects to export, even if they have no CIGI mapping or are not marked as spawnable.

4. Click **Export**.
5. The content library for the selected vehicles will then be exported to the directory listed in ContentLibraryGenerator / ContentExportDir.

6.5.7 Entity List Window

The Entity List Window consists of the following:

- **Entity List** - Lists all the selectable entities in the scene.
- **Entity Info** - Shows information about the selected entity in the Entity List.



To open the Entity List Window, follow these steps:

1. Press backquote (`) / tilde (~) to open the Debug UI.
2. Select **Entity List Window**.

3. In the **Entity List**, use the following controls:

Control	Description
Show Internal	Toggle this to show / hide the internal entities in the **Entity List** window, which are typically non-user created entities.
Show Info	Toggle this to show / hide the **Entity Info** window.
Show Arrows	Toggle this to show / hide transformation arrows at the origin of the currently selected entity.
Arrow Scale	Move slider to adjust size of transformation arrows.
Filter Text	Filter the entities based on the typed string.
Filter Categories	Click the boxes to display only entities selected.

Click an entity in the list to display the information about it in the **Entity Info** UI.

4. In the **Entity Info** dropdown, use the following controls:

<table>
<tr><th>Control</th><th>Description</th></tr>
<tr><td>Attach View</td><td>Expand this to use view-attachment controls for the camera.
Attach - Click this to attach the default camera to the selected entity, when toggled. This can be useful for quickly switching the view to specific entities.
Offset Position - X, Y, Z values of how much to offset the position from the selected entity, when Attach is toggled.
Offset Orientation - X, Y, Z values of how much to offset the orientation from the selected entity, when Attach is toggled.
Smoothing While View Attached
Position Smoothing While Attached - Toggle to enable.
Orientation Smoothing While Attached - Toggle to enable.
Smoothing Time - Use the [-] or [+] to decrement / increment smoothing time.</td></tr>
<tr><td>Entity Info</td><td>Expand this to use Entity Info controls for the camera.
Copy Entity Info to Clipboard - Click this to copy the entity information values to the clipboard.
Entity Info fields - The following fields are populated when an object is selected:
Handle
Entity Type
Mapping
ID
Model Config
Model Type
Model Path
Global Draw Mode - In the dropdown menu, select Normal, Pilot, Gunner, Cargo, Wreck, Geometry.</td></tr>
<tr><td>Hierarchy</td><td>Click to open the information on entity hierarchy:
Parent Entity
Children
NOTE
Option only available if the selected entity is attached to another entity.</td></tr>
</table>

Control	Description
Smoothness	Expand to use and view entity smoothness controls for the camera. • **Use Previous Frame Delta** - Toggle to enable previous frame delta for selected entity. • **Use System Time** - Toggle to use system time for selected entity. • **Position graph** - Visual representation of how smoothly the position of the selected entity changes in time. • **Orientation graph** - Visual representation of how smoothly the position of the selected entity orientation changes in time.
Positions and Movement	Expand to display the information of any of the entities.
Clamping	Expand to display clamping behavior of the selected entity.

6.5.8 Environment Window

The weather environment can be controlled using the **Weather Control** packet (see the online ICD Reference (https://manuals.bisimulations.com/cigiicd/icd_23_2/index.html)) or the **Weather** controls that are available in the Environment Window, accessible through the Debug UI (on page 376).

> **NOTE**
>
> The settings in the Environment Window override the ones sent by the CIGI Weather Control packet and the VBS Host.

Follow these steps:

1. Press backquote (`) / tilde (~) to open the Debug UI.
2. Select **Environment Window**.

3. Set the following parameters (click **Reset To Default Weather from Settings** to reset the parameters to default scenario settings):

Parameter	Description
Current Time of Day	Sets the current time of day. • **Year** (use **+** / **-**, or enter the year in the YYYY format) • **Month** (use the slider to set the month) • **Day** (use the slider to set the day) • **Hour** (use the slider to set the hour) • **Minute** (use the slider to set the minute) • **Second** (use the slider to set the second)
Time of Day Progression Rate	Controls how fast the time of day progresses (use the slider to set it, click Reset to reset to the default time progression, defined in the scenario settings): • 0 - Time stops (click **Freeze** to set it). • 1.0 - Normal time progression. • > 1.0 - Fast-forward time progression.
Environment Simulation Time	This value represents the current environment simulation time in seconds.
Environment Simulation Progression Rate	Controls the time offset of the simulation. This impacts simulation animation such as wind on trees, water motion, particle effects, etc. • 0 - Environment simulation stops (click **Freeze** to set it). • 1.0 - Normal environment simulation progression. • > 1.0 - Fast-forward the environment simulation progression.
Environment Effect Time	This value represents the current environment effect time in seconds.
Environment Effect Progression Rate	Controls the time offset of the effect simulation. This impacts effect simulation animation such as rain, droplets, sun flare, or noise in Thermal Imaging sensor mode. • 0 - Environment simulation stops (click **Freeze** to set it). • 1.0 - Normal environment simulation progression. • > 1.0 - Fast-forward the environment simulation progression.
Environment Adaptation Time	This value represents the current environment adaptation time in seconds.

Parameter	Description
Environment Adaptation Progression Rate	Controls the time offset of the adaptation simulation. This impacts effect simulation animation such as eye adaptation or automatic Thermal Imaging. • 0 - Environment simulation stops (click **Freeze** to set it). • 1.0 - Normal environment simulation progression. • > 1.0 - Fast-forward the environment simulation progression.
Global Precipitation State	Controls the precipitation state for the entire globe. • **Enable Override** - Check to set precipitation rate / uncheck to reset back to default state. • **Type** - Precipitation type. Can be: Rain, Snow, Sleet, Hail. • **Density** - Precipitation density. • **Severity** - Severity of precipitation. • **Max Altitude** - Slider to set the maxim altitude (in meters), at which precipitation is present. • **Puddle Size** - Slider to set puddle size. • **Wetness** - Slider to set wetness of surfaces. • **Snow Amount Ground** - Slider to set amount of snow on ground. • **Snow Amount Trees** - Slider to set amount of snow on trees. • **Snow Amount Buildings** - Slider to set amount of snow on buildings. • **Snow Line Altitude** - Slider to set the altitude where snow begins to appear. This slider is ignored if **Snow Transition Band** has a negative value. • **Snow Transition Band** - Slider to set an altitude offset to **Snow Line Altitude**, where snow reaches full coverage. When set to a negative value, snow is displayed at all altitudes. • **Freeze Water Surfaces** - Check to globally enable frozen water surfaces.
Global Snow Plowing	Controls snow plowing for the entire globe. • **Enable Override** - Check to set snow plowing state / uncheck to reset back to default state. • **Depth** - Adjust snow depth.
Global Fog State	Controls the fog state for the entire globe. • **Enable Override** - Check to set the state / uncheck to reset back to default state. • **Color Preset** - Color including Fog, Sand, Dust, Custom. • **Visibility** - Fog visibility. • **Base Altitude** • **Altitude Scale** • **Custom Fog Color** - RGB fog color.

Parameter	Description
Global Overcast State	Controls the cloud layer properties for the entire globe. • **Enable Override** - Check this to set the cloud layer properties, and uncheck it, to reset back to the default properties. • **Cloud Presets** - Drop down menu options for selecting cloud types include the following: ◦ Cumulus ◦ Low cumulus scattered ◦ Tall cumulus ◦ Low stratus ◦ Medium stratus ◦ High stratus ◦ Cumulus fronts + stratus ◦ Almost overcast light ◦ Almost overcast dense ◦ Dual stratus ◦ Soft overcast ◦ Fly inside **Layers** - Available cloud layers (Layer 0 - 2). Each layer has the following properties: ◦ **Density** - Controls the overall density of the cloud layer. ◦ **Type** - Cloud type: 0 - Cumulus, 1 - Stratus. ◦ **Coverage** - Controls the overall coverage of the cloud layer. ◦ **Height** - Heights above the ellipsoid in meters: • **Height 0** - Height lower bound, where clouds start gaining density. • **Height 1** - Height at which the cloud reaches full density. • **Height 2** - Height at which the cloud layer starts to fall off. • **Height 3** - Height where the layer disappears completely. **NOTE** All heights in Layer 0 should be lower than all heights in Layer 1, and all heights in Layer 1 should be lower than all heights in Layer 2.
Global Sea State	Controls the sea state for the entire globe. • **Enable Override** - Check this to set the sea state, and uncheck it, to reset back to the default state. • **State** - Slider to set the sea state, from calm (0) to maximal storminess (12).
Global Water Color	Controls the water color for the entire globe. • **Enable Override** - Check this to set the water color, and uncheck it, to reset back to the default color. • **Water Color** - RGBA color.

Parameter	Description
Global Wind Speed	Controls the wind speed for the entire globe. • **Wind Speed** - Wind speed (to disable, set to: 0, 0, 0).
Global Lightning	Controls lightning for the entire globe. • **Enable Random Lightning** - Check this box to enable lightning to spawn randomly, and uncheck it to reset to the default state. Once checked, adjust the properties of the lightning with the slider settings below, as required. • **Duration** - Determines how long a lightning strike lasts, measured in seconds. • **Spawn interval** - The amount of time between lightning spawn attempts, measured in seconds. • **Spawn Chance** - The chance of a lightning bolt occurring during every lightning spawn interval, allowing settings from 0-1. • **Min Distance** - The minimum distance lightning can spawn from a viewport, measured in meters. If there are multiple viewports, then a random viewport is selected to spawn lightning every spawn interval. • **Max Distance** - The maximum distance lightning can spawn from a viewport, measured in meters. If there are multiple viewports, then a random viewport is selected to spawn lightning every spawn interval. • **Light Intensity** - Intensity of the light flash from lightning (default set at 2000.0). • **Light Radius** - Radius of the light flash from the lightning.
Color Write Render Features	Controls the parts of the scene that are enabled for rendering. Check any of the options in the list to enable them, and uncheck them to reset to the default state: Sky, Sun, Moon, Stars, Ground, Water, Trees, Bushes, Grasses, Geometry, PointClouds, Objects, Lights, Particles, Clouds, Precipitation.
Depth Write Render Features	Controls the parts of the scene that are enabled for rendering into the depth buffer. Check any of the options in the list to enable them, and uncheck them to reset to the default state: Ground, Water, BiomeTrees, BiomeBushes, BiomeGrasses, Geometry, PointClouds.
Starfield Intensity	Controls the intensity of the starfield.

6.5.9 IG Multichannel Window

The IG Multichannel Window displays the VBS Blue IG client / master information, depending on the VBS Blue IG setup.

Image-6: IG Multichannel Window showing VBS Blue IG client information

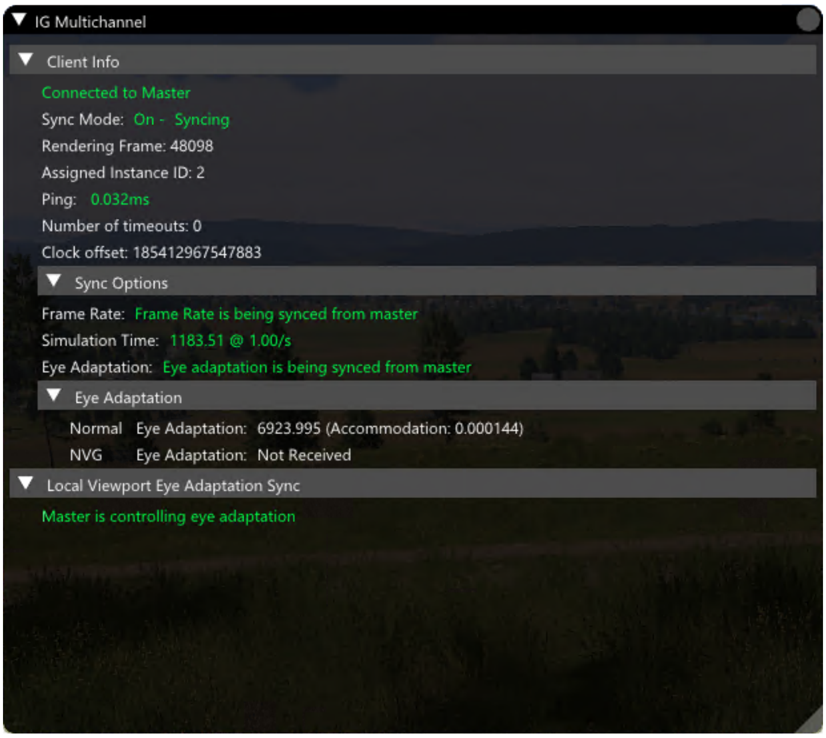
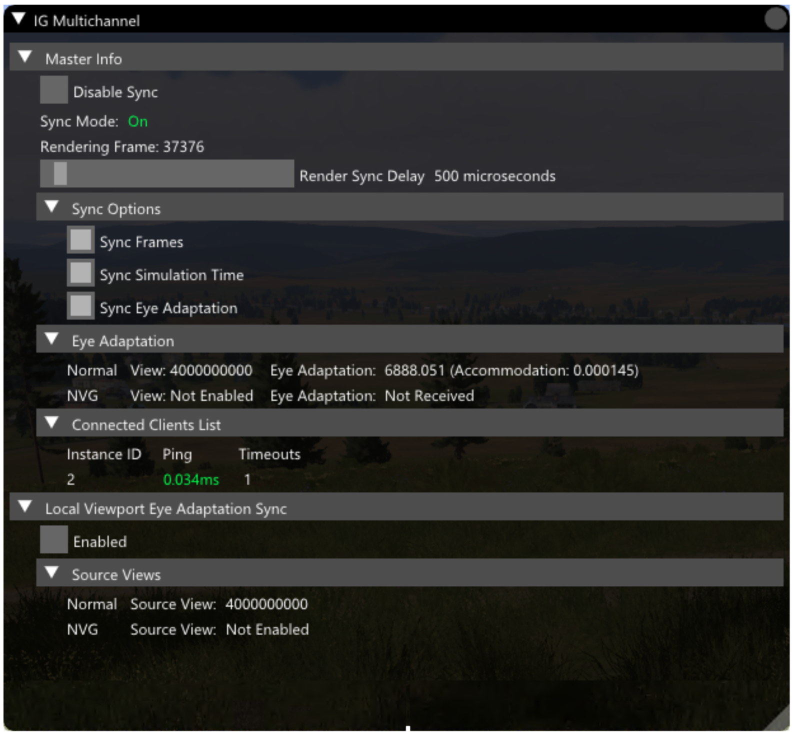


Image-7: IG Multichannel Window showing VBS Blue IG master information



Follow these steps:

1. Press backquote (`) / tilde (~) to access the Debug UI (on page 376).
2. Select **IG Multichannel Window**.
3. Do any of the following:
   - Expand Client Info window (below) to display information for client instances.
     - Expand Sync Options (on the next page) to display sync information.
     - Expand Eye Adaptation (on the next page).
   - Expand Local Viewport Eye Adaptation (on the next page).
   - Expand Master Info window (on the next page) to display additional settings and information for the master instance.
     - Expand Sync Options (on page 401) to display user options for controlling sync.
     - Expand and enable Eye Adaptation (on page 401).
     - Expand Connected Clients List (on page 401).
     - Expand and enable Local Viewport Eye Adaptation Sync (on page 401).

6.5.9.1 Client Info window

Expand to display information for client instances.

Data Field	Description
Sync Mode	Must be **On** for multichannel configurations to run. This can be configured in `IGMultiChannel.xml` (see Views Settings (on page 361)).
Rendering Frame	Current rendering frame.
Assigned Instance ID	ID of the assigned instance.
Ping	Ping reaction time for the connection.
Number of timeouts	Number of connection timeouts that have occurred.
Clock offset	Clock offset from the clock on the VBS Blue IG master.

6.5.9.1.1 Sync Options

Expand Sync Options to display sync information below.

Data Field	Description
Frame Rate	Shows sync status of frame rate.
Simulation Time	Shows current Simulation Time from the Master.
Eye Adaptation	Shows sync status of eye adaptation.

6.5.9.1.2 Eye Adaptation

Expand to display data values below.

Data Field	Description
Normal Eye Adaptation	If Eye Adaptation enabled, displays sync data values for normal eye adaptation between viewports on this VBS Blue IG instance.
NVG Eye Adaptation	If Eye Adaptation enabled, displays sync data for NVG eye adaptation between viewports on this VBS Blue IG instance.

6.5.9.1.3 Local Viewport Eye Adaptation

Expand to display Local Viewport Eye Adaptation Sync status.

6.5.9.2 Master Info window

Expand to display additional settings and information for the master instance.

Data Field	Description
Disable Sync	Disables multichannel synchronization (see Views Settings (on page 361)).
Sync Mode	Must be **On** for multichannel configurations to run. This can be configured in `IGMultiChannel.xml` (see Views Settings (on page 361)).
Rendering Frame	Adjust current rendering frame.
Render Sync Delay	Synchronization delay for rendering, between VBS Blue IG master and clients.

6.5.9.2.1 Sync Options

Expand **Sync Options** to display user options for controlling sync.

Data Field	Description
Sync Frames	Enable sync frame rate.
Sync Simulation Time	Enable sync Simulation Time.
Sync Eye Adaptation	Enable sync eye adaptation.

6.5.9.2.2 Eye Adaptation

If enabled, data values below are available.

Data Field	Description
Normal	If Eye Adaptation enabled, displays sync data values for normal eye adaptation between viewports on this VBS Blue IG instance.
NVG	If Eye Adaptation enabled, displays sync data for NVG eye adaptation between viewports on this VBS Blue IG instance.

6.5.9.2.3 Connected Clients List

If clients are connected, data values for clients are listed.

6.5.9.2.4 Local Viewport Eye Adaptation Sync

Expand **Local Viewport Eye Adaptation Sync**. If enabled, data values below are available in the **Source Views** drop-down menu.

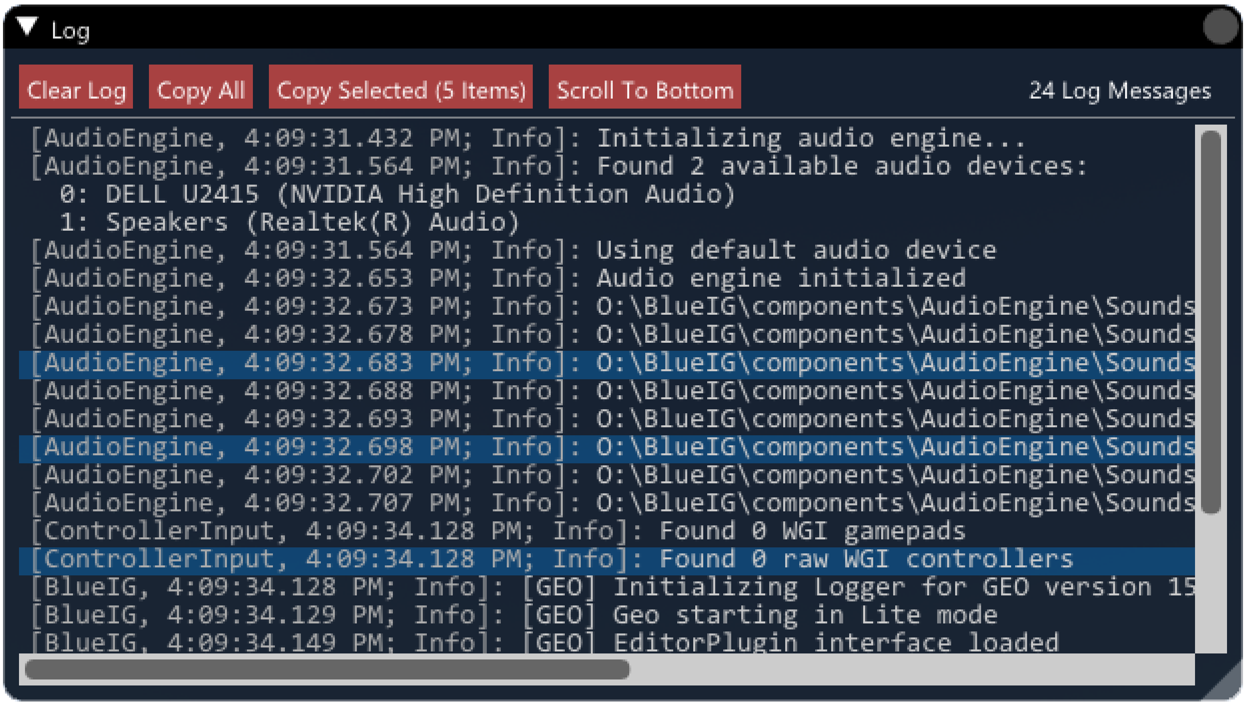
Data Field	Description
Normal	If Eye Adaptation enabled, displays sync data values for normal eye adaptation between viewports on this VBS Blue IG instance.
NVG	If Eye Adaptation enabled, displays sync data for NVG eye adaptation between viewports on this VBS Blue IG instance.

6.5.10 Log Window

The log window outputs status information about VBS Blue IG.

Besides the functionality within the Log Window (above) described below, it can also be controlled by customizing output in VBS Blue IG Settings (on page 201).

> **NOTE**
>
> Changes to the VBS Blue IG Settings window in Log Settings (on page 257) may require a restart of the application in order for Debug UI Log Window changes to take effect.



Follow these steps:

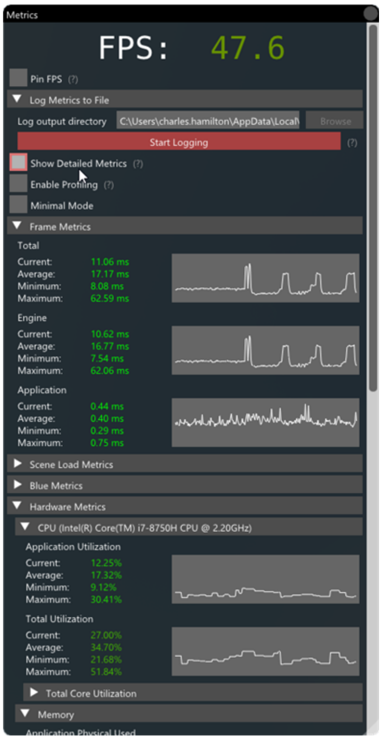
1. Press backquote (`) / tilde (~) to access the Debug UI (on page 376).
2. Select **Log Window**.

3. Choose from the following options:

Parameters Section	Description
Clear Log	Clears all log entries.
Copy All	Copies the entire log to the clipboard.
Copy Selected Items	Copies selected items to the clipboard. Selecting individual or multiple entries is done using **CTRL + Click** or **SHIFT + Click**, respectively.
Scroll To Bottom	This function becomes visible only when the last entry is not visible and allows scrolling the log to display the last logged entry.

6.5.11 Metrics

The Metrics window displays a realtime view of FPS data of a VBS Blue IG instance.



Follow these steps:

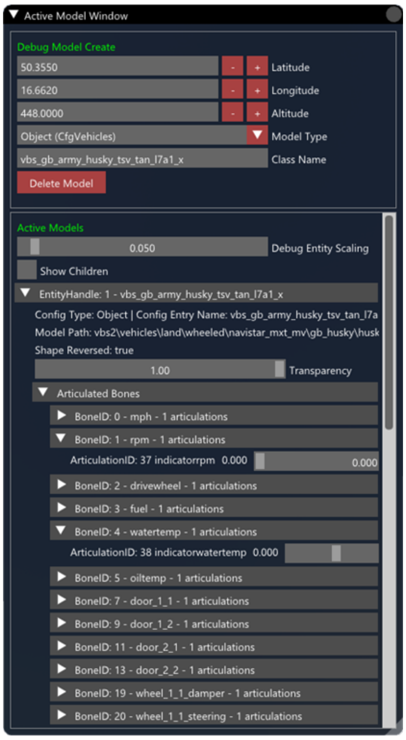
1. Press backquote (`) / tilde (~) to access the Debug UI (on page 376).
2. Select **Metrics**.
3. Expand or select any of the following:

Parameters	Description
Pin FPS	Pins the FPS to the center of the window.
Log Metrics to File	Click the dropdown show the following: • **Start Logging** - Begins logging metrics to designated file • **Log Output Directory** - Target directory for log file. Once logging is started, you can also annotate the file by clicking **Add**.

<table>
<tr><th>Parameters</th><th>Description</th></tr>
<tr><td>Show Detailed Metrics</td><td>When enabled, shows detailed application and system metrics. This affects the performance of the application and should be disabled unless necessary. Optional metrics shown include the following:
Scene Load Metrics
Blue Metrics
Hardware Metrics</td></tr>
<tr><td>Enable Profiling</td><td>Enables and opens Microprofile, which is used to perform advanced inspection of frame timing. When active, the following options are available:
Enable Frequent Scopes - Enabling frequent profiling scopes further effect performance, but provide further detail. Disable unless necessary.
Open Browser - Opens browser window with multiple options for displaying real-time data and metrics.
NOTE
This option affects the performance of the application and should be disabled unless necessary.
For more information, see Create Profile Captures (on page 464).</td></tr>
<tr><td>Minimal Mode</td><td>Enables transparent overlay of metrics onto scenario.</td></tr>
<tr><td>Frame Metrics</td><td>Click the dropdown to show realtime FPS.</td></tr>
<tr><td>Scene Load Metrics</td><td>Total Load Percent - Shows summary scene load including Current, Average, Minimum, and Maximum.
Individual Metrics - Shows load metrics according to individual scene load for Ground, Water, Clouds, PointCloud, Biome, Geometry, Objects</td></tr>
<tr><td>Blue Metrics</td><td>Shows metrics for number of Triangles, Primitives, and Additional Blue Engine queries.</td></tr>
<tr><td>Hardware Metrics</td><td>CPU - Shows summary hardware metrics for Application Utilization and Total Utilization.
Total Core Utilization - Shows individual utilization by each CPU core.
Memory - Shows memory metrics for Application Physical Used, Application Virtual Used, Total Physical Used, Total Virtual Used.
GPU - Shows GPU metrics for Application Utilization, Total Utilization, DXGI VRAM Used, Application VRAM Used and Total VRAM Used.</td></tr>
</table>

6.5.12 Model Controller

The Model Controller window allows you to create, delete and modify placed models.



Follow these steps:

1. Press backquote (`) / tilde (~) to access the Debug UI (on page 376).
2. Select **Model Controller**.
3. Choose from the following options:
4. Adjust the **Model Controller** settings as desired. For more information, see Model Controller Debug Settings (below).

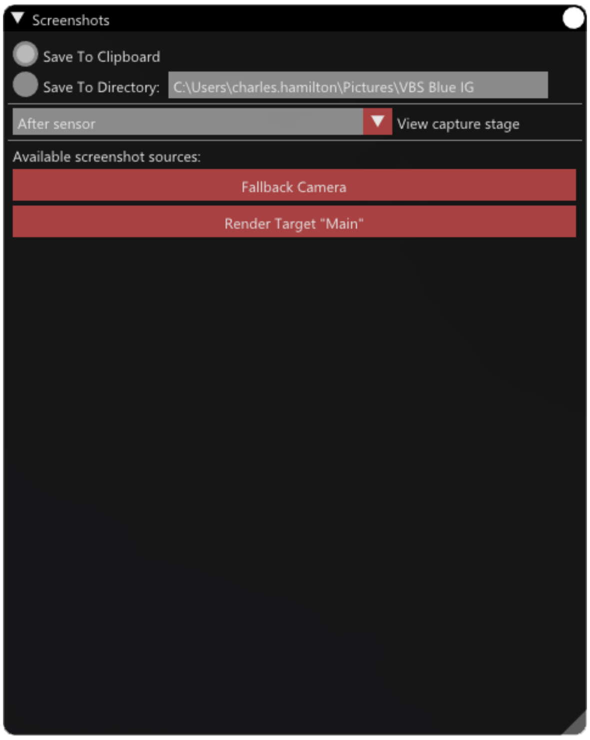
6.5.12.1 Model Controller Debug Settings

Parameters	Description
Create / Delete Model	Select this button to create or delete any model (given a class name) at a specific coordinate.

Parameters	Description
Active Models	Once a model is created, the options below become available: • **Show Children** - Toggling this option adds child entities to the active model list. Child entities are typically mounted machine guns and other things that can be attached to vehicles and lifeforms. • **Transparency** - A slider that adjusts the transparency level of a model. • **Articulated Bones** - A list of bones on the model that can be animated. Each articulated bone has a slider that adjusts the position of the bone. • **Articulated Sources** - Similar to articulated bones, a list of animations for the model. Each articulation source has a slider that can adjust the articulation. • **Selections** - A list of the selections of the model. Each selection is a part of the model and can be toggled on or off. • **Proxies** - Similar to children, models have 'proxies'. A proxy is just another model that is attached to the selected model. The 'Proxies' list shows the entity handle for each proxy.

6.5.13 Screenshots

The Screenshots window allows you to capture screen images of the current instance of VBS Blue IG.



Follow these steps:

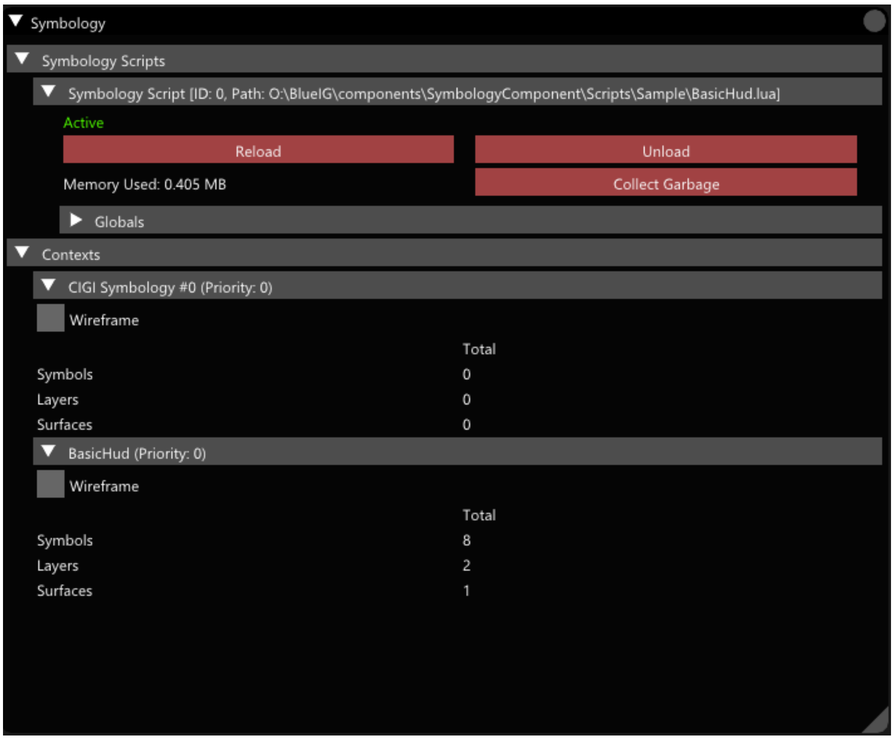
1. Press backquote (`) / tilde (~) to access the Debug UI (on page 376).
2. Select **Screenshots**.
3. Choose from the following options:

Parameters Section	Description
Save to Clipboard	Select to save screenshot to the clipboard.
Save to Directory	Select to save screenshot to a specified directory.
View Capture Stage	Click dropdown to select from the following options: • After sensor • After render

4. To capture an image of the IG View window only, click **Fallback Camera**.
5. To capture an image of the main window along with any open application windows, click **Render Target "Main"**.

6.5.14 Symbology

The Symbology window displays debugging information about CIGI-based Symbology.



Follow these steps:

1. Press backquote (`) / tilde (~) to access the Debug UI (on page 376).
2. Select **Symbology**.
3. Expand Symbology Scripts to display active Symbology Lua Scripts (on page 165):

Item	Description
Active / Unloaded	The state of the script. If the script encountered an error, the last error will also display (see the Log for additional information).
Reload	Reloads the specified Lua script, allowing for any script edits to be applied (available when a script is loaded).
Unload	Unload and shutdown the specified Lua script (available when a script is loaded).
Collect Garbage	Runs the Lua garbage collector (available when a script is loaded).
Globals	Lists all global Lua functions and variables (available when a script is loaded).
Load	Load and initialize the specified Lua script (available when a script is unloaded).

4. Expand Contexts to display active Symbology contexts:

Item	Description
Wireframe	Check this option to visualize all Symbology in this context with wireframe rendering.
Total	This section displays counts of all currently created Symbols, Layers, and Surfaces in the context.

6.5.15 VBS External Networking

VBS External Networking is the VBS Interop client networking window, providing status information about the networking session. In order for a Interop session to work, both TCP and Multicast must be connected.

The VBS External Networking UI affects the configuration in `VBSExternalNetworking.xml`. However, settings in this menu only perform during runtime. For more information about using persistent settings, see VBS External Networking Settings (on page 324).

Follow these steps:

1. Press backquote ` / tilde **~** to access the Debug UI (on page 376).
2. Select **VBS External Networking**.
3. Set / observe the following:
4. Adjust the **VBS External Networking** settings as desired. For more information, see VBS External Networking Debug Settings (on the next page).

6.5.15.1 VBS External Networking Debug Settings

Setting / Data Field	Description
Connection State	Click the drop-down arrow for the following: • **Protocol version** - Indicates version of the VBSExternalNetworking component's protocol. • **Client ID** - A unique client ID assigned by the host to this VBS Blue IG instance. • **Host Connection** - Indicates current state of TCP connection with a host: ◦ Disconnected (details) Reconnection attempt in countdown ◦ Disconnected (details) Reconnecting... ◦ Waiting for Handshake ◦ Connected Host type, host protocol version, connection duration • **Host Multicast Presence - Time Since Last 3s Heartbeat** - A collection of hosts broadcasting their presence (via a heartbeat message) to the same multicast socket (i.e. address:port) as this VBS Blue IG instance is listening to. Each record indicates a host's TCP address and port, its type, protocol version, and time since the last received heartbeat.

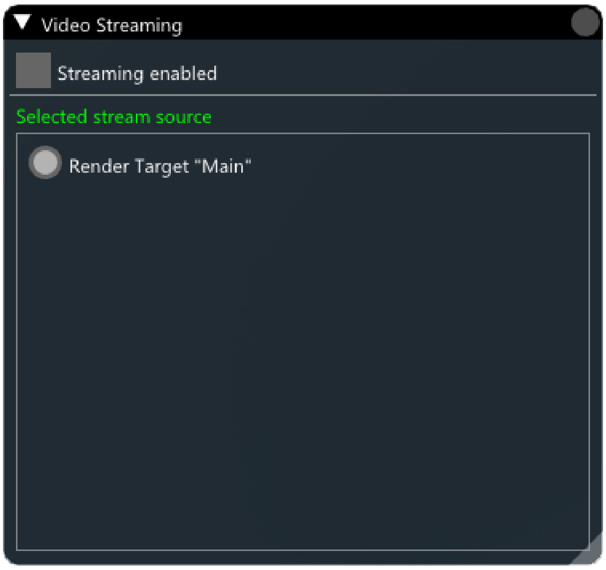
Setting / Data Field	Description

Network Mode	Indicates networking status: initialized / not initialized.
	Click the drop-down arrow to adjust or modify the following:
	• **Live / Playback** - Network incoming traffic can be recorded into a file and played back with an alternative **playback** network handler. Select one of these options:
	  ◦ **Live** - Enable recording of incoming network traffic.
	  ◦ **Playback** - Switch VBS Blue IG to a playback mode.
	For detailed steps on using this feature, see Recording and Playback of VBS External Networking traffic (on page 445).
	• **Reliable Socket Settings**:
	  ◦ **Remote Host Port** - Send To port number (use **+** / **-**, or enter the port number).
	  ◦ **Remote Host Address** - Host address to enter.
	• **Multicast Settings** - Adjust the Multicast settings, as required:
	  ◦ **Multicast Port** - Multicast port number (use + / -, or enter the port number).
	  ◦ **Multicast Address** - Multicast address (enter the address).
	  ◦ **Multicast NIC** - Local Network Interface Controller (NIC) to use for the multicast communication (enter the address).
	• **UDP Settings** - Adjust the UDP settings, as required:
	  ◦ **UDP Host Port** - Host UDP port number to connect to (use **+** / **-**, or enter the port number).
	  ◦ **UDP NIC** - Local Network Interface Controller (NIC) to use for the UDP communication (enter the address).
	• **Reset Network** - Resets the networking with the currently configured settings.
	• **Incoming Net Stats** - Other data fields include the following:
	  ◦ Packets in queue
	  ◦ Avg. processed packets size
	  ◦ Avg. valid messages per packet
	  ◦ Avg. valid messages size
	  ◦ Time since last packet
	  ◦ Total packets received
	  ◦ Total packets processed
	  ◦ Avg. receive to process latency
	  ◦ My sampled frame rate

<table>
<tr><th>Setting / Data Field</th><th>Description</th></tr>
<tr><td></td><td>Message Stats - Contains controls for activating network messages statistics collection at the given endpoint (VBS host or VBS Blue IG client), for resetting collected stats, and for exporting stats to <code>csv</code> files at the given target directory.
Activate - Click to enable stats collection.
Reset - Resets collected stats.
Export - Click to export collected stats into designated file.
Messages Recorder - Contains controls for activating, resetting and setting a target directory for saving messages received from a VBS Host.
Activate - Click to enable messages recorder.
Reset - Resets collected stats.</td></tr>
<tr><td>VBS Host Listeners</td><td>Click the drop-down arrow to see a list of active listeners related to the VBS Host/ IG interconnection events and handling messages received from the connected VBS Host.</td></tr>
</table>

6.5.16 Video Streaming

The Video Streaming window allows you to enable streams of a scenario.



Follow these steps:

1. Press backquote (`) / tilde (~) to access the Debug UI (on page 376).
2. Select **Video Streaming**.
3. Choose from the following options:

Parameters Section	Description
Streaming enabled	Click to enable stream.
Selected Stream Source	Choose a render target or view ID to stream. By default, the Render Target "Main" is available until additional render targets and views are created.

4. Modify additional stream parameters within the Settings UI. For more information, see Streaming Settings (on page 314).

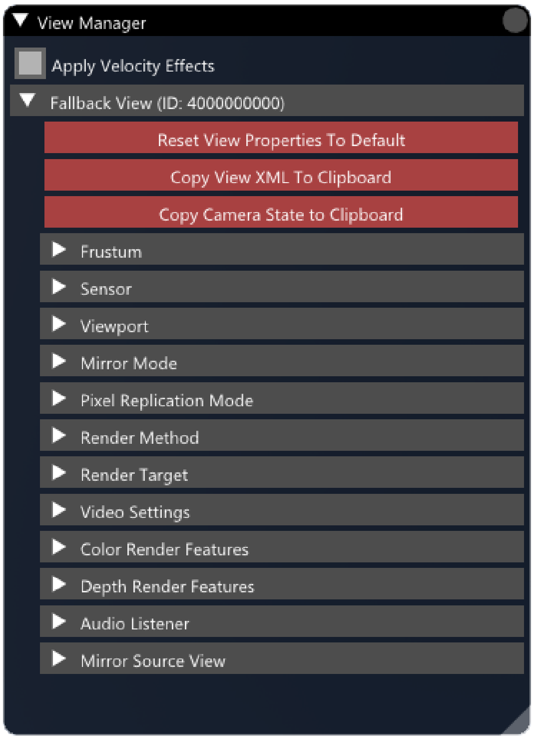
6.5.17 View Manager

The View Manager allows you to view and configure all the currently registered views in the current instance of VBS Blue IG.

Views and Render Targets must be created through some other means (CIGI, VBS4, or VBS IG SDK). However, once created, their properties can be dynamically configured using the View Manager, which is useful for updating parameters at runtime to find the ideal settings.

For repeated use, these settings can be saved in the view configuration file found in the following folder:

`\IG_Installation\Components\BlueIGViewSystems\Config\DefaultViewConfig.xml`.



Follow these steps:

1. Press backquote ` / tilde ~ to access the Debug UI (on page 376).
2. Select **View Manager**.
3. Click **Apply Velocity Effects** for all configured views.

   When enabled, simulation effects (such as rain direction) will respond to the velocity of the camera. For the setting to take effect, the camera must be attached to a moving entity, like a lifeform or a vehicle.
4. Select a view and expand to see its available options.
5. To reset the view properties to default settings (as defined in `DefaultViewConfig.xml`), click **Reset View Properties To Default**.

6. To copy the contents of `DefaultViewConfig.xml` to the clipboard, modified using the View Manager UI, click **Copy View XML To Clipboard**.
7. To copy the camera state in XML format to the clipboard, click **Copy Camera State To Clipboard**.
8. Adjust the **View Manager** settings as desired. For more information, see View Manager Settings (below).
9. Save the XML file.
10. Restart VBS Blue IG.

6.5.17.1 View Manager Settings

Control the view using the View Manager Settings.

Parameters Section	Description
Frustum	Frustum controls. • **Near**, **Far**, **Left**, **Right**, **Top**, **Bottom** - Frustum attributes. Use the sliders to set. • **Projection** - Frustum projection type. Select from the following dropdown options: ◦ Perspective ◦ Orthographic
Sensor	Controls for the sensor, used to render the view. For more information, see Sensor Types (on page 419).
Viewport	Controls the viewport for the selected view. • **Left**, **Top**, **Right**, **Bottom** - Viewport side attributes. Use the sliders to set. • **Layer** - Viewport layer, use **+** / **-**, or enter the layer number. **NOTE** The viewport can only be modified if the view is rendered to a render target that is separate from the main render target.
Mirror Mode	Mirrors the View depending on the selected parameter: • None • Horizontal • Vertical • HorizontalAndVertical

<table>
<tr><th>Parameters Section</th><th>Description</th></tr>
<tr><td>Pixel Replication Model</td><td>Duplicates pixels of a View depending on the selected parameter:
None
1x2 - Every vertical pixel is duplicated twice. The View will appear vertically stretched.
2x1 - Every horizontal pixel is duplicated twice. The View will appear horizontally stretched.
2x2 - Every horizontal and vertical pixel is duplicated twice. The View will have a 2x digital zoom, though it may not be sharply focused.</td></tr>
<tr><td>Render Method</td><td>Controls the rendering method to use for the view. In the Method dropdown, select from the following options:
None
Screen
Full screen
RTT
Fallback
MultiViewParent
MultiView</td></tr>
<tr><td>Render Target</td><td>Controls the render target that the view is rendered to.
Render Target Handle - A unique identifier assigned to the Render Target upon its creation.
Render Target ID - Render target ID, use + / -, or enter the ID number.
NOTE
The render target must first be created in order to be able to switch the view to render to another render target.</td></tr>
<tr><td>Video Settings</td><td>Defines viewport-specific video settings. Video settings specified here overrides the corresponding global video setting for a specified view. If not defined (default), the view uses the global video settings.
Select the Show View Video Setting Overrides button to open a settings window for the specified view.
Parameters are specified as child VideoSetting nodes. Each VideoSetting node consists of 2 attributes:
@Name - The full name of the video setting identifier.
@Value - The value for the video setting. The value type must match the parameter type.</td></tr>
</table>

<table>
<tr><th>Parameters Section</th><th>Description</th></tr>
<tr><td>Color Render Features</td><td>An array of features enabled for rendering to the color buffer. If not defined (default), the view applies the render features from the global Environment settings. If defined but empty, the view does not render anything to the color buffer.
Click the Override Global Render Features to enable button to enable the following ColorRenderFeature values which can be optionally overridden:
Sky
Sun
Moon
Stars
Ground
Water
BiomeTrees
BiomeBushes
BiomeGrass
Geometry
PointClouds
Objects
Lights
Particles
Clouds
Precipitation</td></tr>
<tr><td>Depth Render Features</td><td>An array of features enabled for rendering to the depth buffer. If not defined (default), the view applies the render features from the global Environment settings. If defined but empty, the view does not render anything to the depth buffer.
Click the Override Global Render Features to enable button to enable the following DepthRenderFeature values which can be optionally overridden:
Ground
Water
BiomeTrees
BiomeBushes
BiomeGrass
Geometry
PointClouds</td></tr>
<tr><td>Audio Listener</td><td>Sets the view as an audio listener. Select a Selection Method from the drop-down menu:
Auto - The audio listener is determined automatically from the topmost view.
Forced - The selected view is forced to be the listener.</td></tr>
<tr><td>Mirror Source View</td><td>Sets the mirror source view. Select a Selection Method from the drop-down menu:
Auto - Set the first available mirror source view.
Forced - The view should be forced for given mirror.</td></tr>
</table>

6.5.17.1.1 Sensor Types

<table>
<tr><th>Type</th><th>Parameter</th></tr>
<tr><td>Blankscreen</td><td>Color - Adjusts the color values of the blank screen:
Red
Green
Blue
Alpha</td></tr>
<tr><td>Classification</td><td>No parameters available for this sensor.</td></tr>
<tr><td>Disable</td><td>No parameters available for this sensor.</td></tr>
</table>

Type	Parameter
MRT Debug	Adjust the following parameters: • **Render Layer** - Choose the MRT layer to render: ◦ Combined ◦ Opaque • **Zoom Region** - Choose the MRT zoom region: ◦ None ◦ Albedo ◦ Light ◦ Near Infrared ◦ Atmosphere ◦ Color Image ◦ Normals ◦ Internal Heat ◦ Sun Factor ◦ Materials ◦ Eye Accommodation ◦ Classification ◦ Motion ◦ Atmosphere Transmittance

<table>
<tr><th>Type</th><th>Parameter</th></tr>
<tr><td>Night vision</td><td>
Select from the following parameters:

Auto Eye Accommodation - Uncheck this to use the Eye Accommodation slider settings option. When checked, the Auto Eye Accommodation Parameters drop-down becomes available, and includes the following settings:

Min Luminance
Max Luminance
Time Constant Rise
Time Constant Drop
Target Brightness
Focus Auto Center
Focus Centre X
Focus Centre Y
Focus Radius
Focus Dazzling
Focus Dazzling Radius

Night Vision - Adjust the following, as needed:

Gain
Gain Pow
Blooming
Blooming POW
Center Intensity

Tonemap - Select from the following options:

Exposure
Contrast
Saturation
Vibrance
ToneMapper Type
Cutoff
Temperature Shadow
Temperature Highlight
Tint Shadow
Tint Highlight
Display Luminance
Scotopic Range
Scotopic Desaturation
Scotopic BlueShift
Scotopic Darkening

Bloom - Parameters for bloom effect. Select from the following options:

Intensity
Spread Multiplier
Chromatic Spread Multiplier

Display - Simulate display-like brightness on RTT surfaces. When enabled, the following options become available:

Environment Luminance
Backlight Luminance
Backlight Color

Red
Green
Blue

Digital Zoom - Using the slider, adjust the digital zoom process parameters. Models digital zoom up-sampling with respect to the render surface. Rescales the input from the center based on factor. Values less than or equal to 1 have no effect.

</td></tr>
</table>

Type	Parameter
Normal	Select from the following parameters: • **Auto Eye Accommodation** - Uncheck this to use the **Eye Accommodation** slider settings option. When checked, the **Auto Eye Accommodation** Parameters drop-down becomes available, and includes the following settings: ◦ Min Luminance ◦ Max Luminance ◦ Time Constant Rise ◦ Time Constant Drop ◦ Target Brightness ◦ Focus Auto Center ◦ Focus Centre X ◦ Focus Centre Y ◦ Focus Radius ◦ Focus Dazzling ◦ Focus Dazzling Radius • **Color Vision** ◦ **Noise** - Adjust the following, as needed: • Dynamic • Static • Size • Hot • Hot Threshold ◦ **Monochromatic** - If enabled, noise will be grayscale. Otherwise, RGB noise is produced. ◦ **Blurring** - Adjust image blurring. • **Tonemap** - Select from the following options: ◦ Exposure ◦ Contrast ◦ Saturation ◦ Vibrance ◦ ToneMapper Type ◦ Cutoff ◦ Temperature Shadow ◦ Temperature Highlight ◦ Tint Shadow ◦ Tint Highlight ◦ Display Luminance ◦ Scotopic Range ◦ Scotopic Desaturation ◦ Scotopic BlueShift ◦ Scotopic Darkening • **Bloom** - Parameters for bloom effect. Select from the following options: ◦ Intensity ◦ Spread Multiplier ◦ Chromatic Spread Multiplier • **Depth of Field** - Depth of field post-process parameters. When selected, the following options become available: ◦ Near Beg ◦ Near End ◦ Near Blur Radius ◦ Far Beg ◦ Far End ◦ Far Blur Radius

<table>
<tr><th>Type</th><th>Parameter</th></tr>
<tr><td></td><td>

Display - Simulate display-like brightness on RTT surfaces. When enabled, the following options become available:

Environment Luminance
Backlight Luminance
Backlight Color
Red
Green
Blue

Digital Zoom - Using the slider, adjust the digital zoom process parameters. Models digital zoom up-sampling with respect to the render surface. Rescales the input from the center based on factor. Values less than or equal to 1 have no effect.
Color Correction - Using the sliders, adjust the following color correction parameters:

Colorization - Color used for colorization of final image.
Blending - Color blended to the final image.

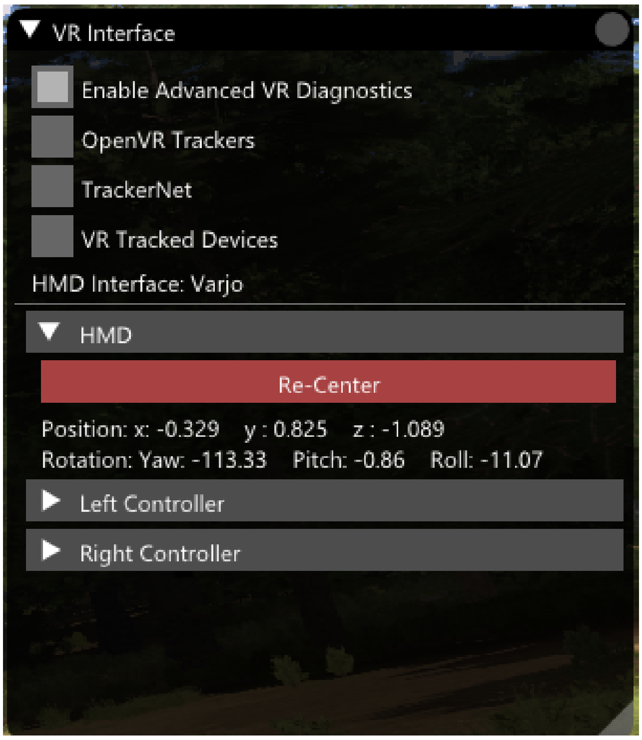
</td></tr>
<tr><td>SAR</td><td>
Simulation - Select from the following options:
Auto Calculate Fov
Jitter Amount
SAR - Select from the following options:
Focus Width
Transmitter Power
Sidelobe Threshold
</td></tr>
</table>

Type	Parameter

Thermal Imaging	• **Thermal Imaging** - Select from the following options:
	◦ Dynamic Range Mode
	◦ Dynamic Range
	◦ Dynamic Range Multiplier
	◦ Black Level Mode
	◦ Black Level
	◦ Black Level Offset
	◦ Noise Dynamic
	◦ Noise Static
	◦ Blur
	◦ Blur Edge
	◦ Lut Index
	◦ Sensor Temperature Minimum
	◦ Sensor Temperature Maximum
	◦ Sensor Wavelength Minimum
	◦ Sensor Wavelength Maximum
	◦ Scan Line State
	• **Display** - Simulate display-like brightness on RTT surfaces. When enabled, the following options become available:
	◦ **Environment Luminance**
	◦ **Backlight Luminance**
	◦ Backlight Color
	  • Red
	  • Green
	  • Blue
	• **Depth of Field** - Depth of field post-process parameters. When selected, the following options become available:
	◦ Near Beg
	◦ Near End
	◦ Near Blur Radius
	◦ Far Beg
	◦ Far End
	◦ Far Blur Radius
	• **Digital Zoom** - Using the slider, adjust the digital zoom process parameters. Models digital zoom up-sampling with respect to the render surface. Rescales the input from the center based on factor. Values less than or equal to 1 have no effect.

6.5.18 VR Interface

The VR Interface window provides information and diagnostic tools for head-mounted displays (HMDs) and VR-related devices supported by VBS Blue IG.



Follow these steps:

1. Press backquote (`) / tilde (~) to access the Debug UI (on page 376).
2. Select **VR Interface**.

3. Click the checkbox **Enable Advanced VR Diagnostics** to add additional diagnostic options.
4. Set / observe the following:

Setting / Data Field	Description
OpenVR Trackers	If the tracker is currently active and connected, it should be visible on this dialog. **NOTE** If the tracker is not visible, ensure that it is turned on and paired to SteamVR.
TrackerNet	Shows connection status of TrackerNet and network VR devices currently visible.
VR Tracked Devices	Intended for SDK debugging. Displays mapping and connection status of **VRTrackedDevices** enumerations for VR API.
HMD Interface	Features and values for this function appear when an HMD is connected. Click **Re-Center** to re-center the HMD, as required. You may also use the keyboard shortcut **LCtrl** + **LShift**.

For more information on setting parameters for VR devices, see VR / Networking (on page 370).

6.5.19 Warping Status

The Warping Status menu gives you access to debugging values related to Warping and Blending.

The Warping Status is divided into sections:

- Scalable Warping (below) - Scalable Warping uses the Scalable Display Manager warp and blend solution.
- DomeProjection Warping (on the next page) - DomeProjection Warping uses the DomeProjection ProjectionTools warp and blend solution.
- Generic Warping (on page 429) - Generic Warping represents view overrides to provide a generic method of integrating with external warp and blend solutions.

> **NOTE**
> Only one warping solution can be enabled at a time.

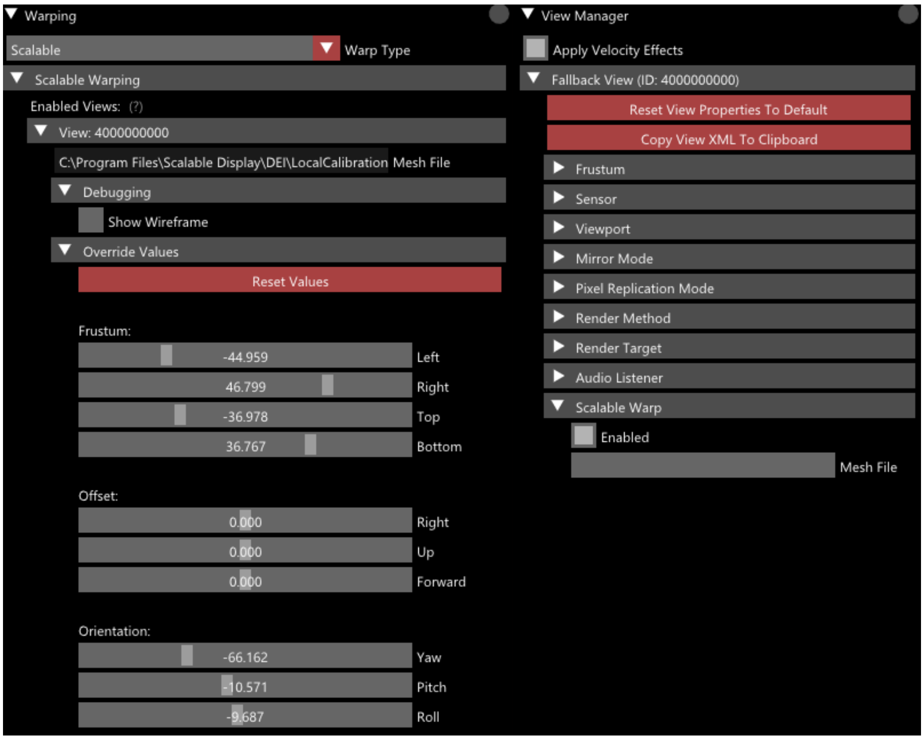
6.5.19.1 Scalable Warping

Scalable Warping is used with the Scalable Display Manager.

Follow these steps:

1. Confirm that VBS Blue IG is started with the `-scalable` startup parameter (for more information, see Scalable Warping Setup (on page 140)).
2. Press backquote **`** / tilde **~** to access the Debug UI (on page 376).
3. Select **Warping Status** and expand **Scalable Warping** in the **Warping Status Menu**.

4. When a view is configured to use Scalable warping, it will display in the **Warping Status** window. The view's configuration is available in the **ViewManager** Debug UI window/



5. Observe what Scalable mesh file (*.ol) (see Scalable Warping Setup) is currently being used.
6. Select Show Wireframe to show a wireframe view of the mesh that Scalable Warping generates.
7. Override any frustum, position offset, or orientation offset to adjust the view if necessary.

6.5.19.2 DomeProjection Warping

DomeProjection Warping is used with DomeProjection ProjectionTools.

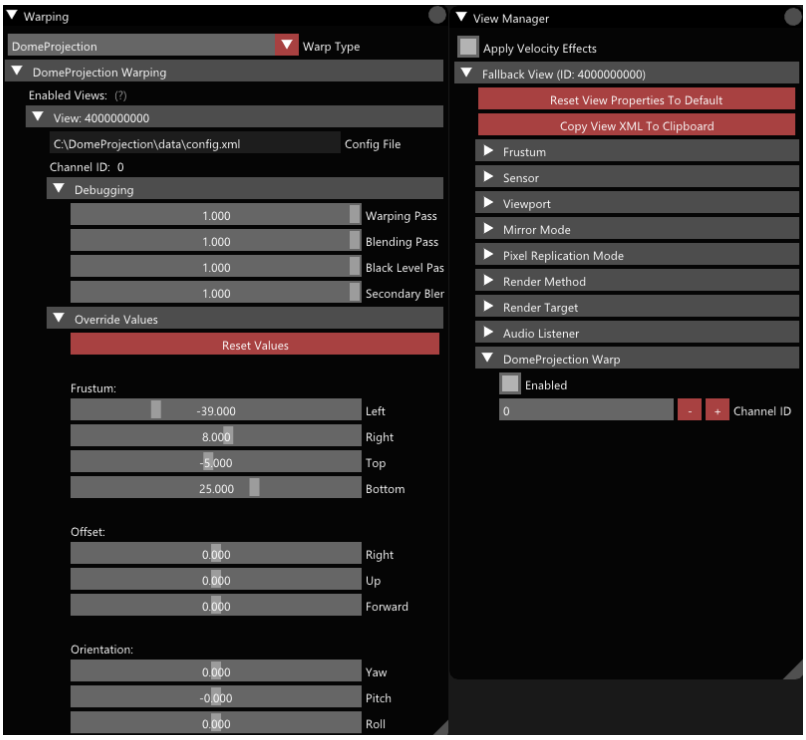
Follow these steps:

1. Confirm that VBS Blue IG is started with the `-domeprojection` startup parameter. For more information, see DomeProjection Warping Setup (on page 137).

   Alternatively, **DomeProjection** can be selected and enabled in the **Warping Status** menu, under **Warp Type**.

2. Press backquote ` / tilde **~** to access the Debug UI (on page 376).
3. Select **Warping Status** and expand **DomeProjection Warping** in the **Warping Status Menu**.

4. When a view is configured to use DomeProjection warping, it will display in the **Warping Status** window. The view's configuration is available in the **ViewManager** Debug UI window.



5. Observe what Config file and channel ID (see DomeProjection Warping Setup (on page 137)) is currently being used.
6. Adjust the effect of each pass. This can be used to visualize the effect each pass provides to the final image.
7. Override any frustum, position offset, or orientation offset to adjust the view if necessary.

6.5.19.3 Generic Warping

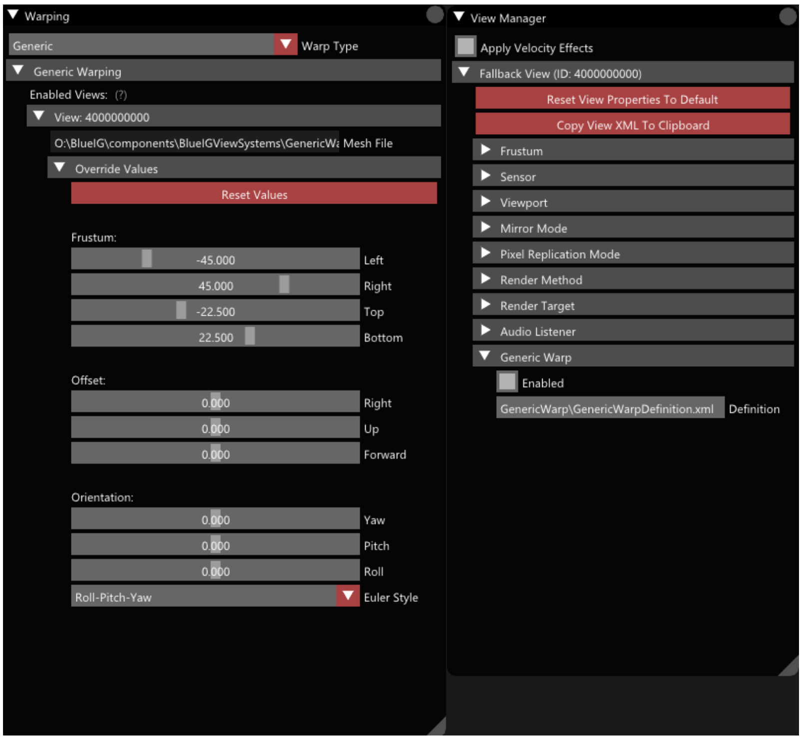
With Generic warping, you can apply overrides to a view's frustum, as well as apply additional position and orientation offsets to the view. This allows for integration with external warping and blending solutions.

Follow these steps:

1. Confirm that VBS Blue IG is started with the `-warp` startup parameter. For more information, see Generic Warping Setup (on page 133)).

   Alternatively, **Generic** can be selected and enabled in the **Warping Status** menu, under **Warp Type**.
2. Press backquote (`) / tilde (~) to access the Debug UI (on page 376).
3. Select **Warping Status** and expand **View Warping** in the **Warping Status Menu**.

4. When a view is configured to use DomeProjection warping, it will display in the **Warping Status** window. The view's configuration is available in the **ViewManager** Debug UI window.

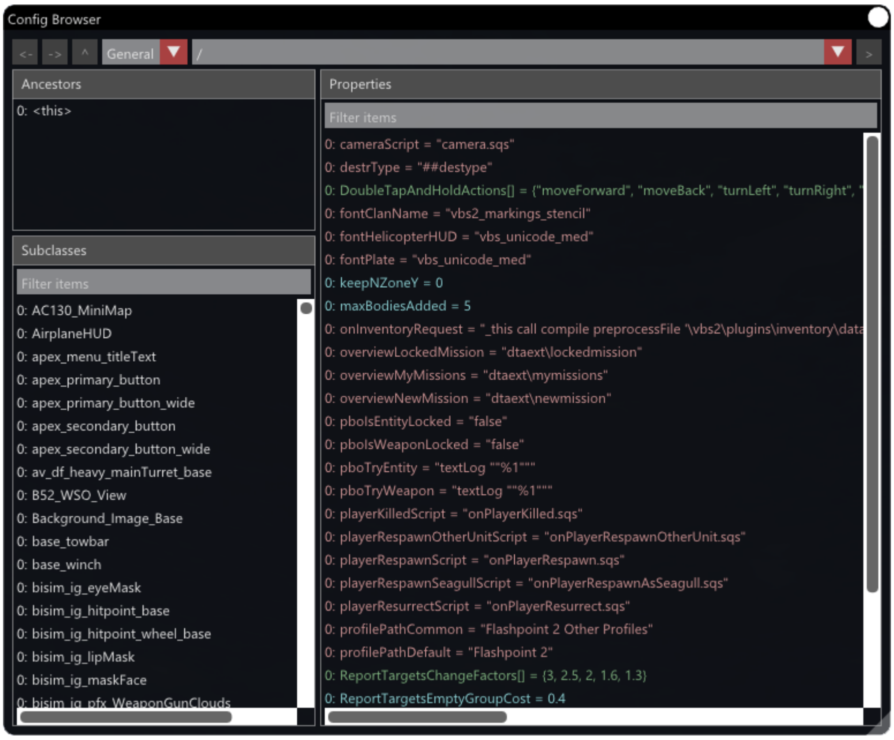


5. Observe what warp definition file (see Generic Warping Setup (on page 133)) is currently being used.

6. Override any frustum, position offset, or orientation offset to adjust the view if necessary.

6.5.20 Config Browser

The Config Browser window in the Debug UI provides views of the configuration of any model, illustrate its class inheritance, subclasses structure, and parameter definitions providing insight into how the model is built.



Follow these steps:

1. Press backquote (`) / tilde (~) to access the Debug UI (on page 376).
2. Select **Config Browser**.
3. Select any of the following panels:

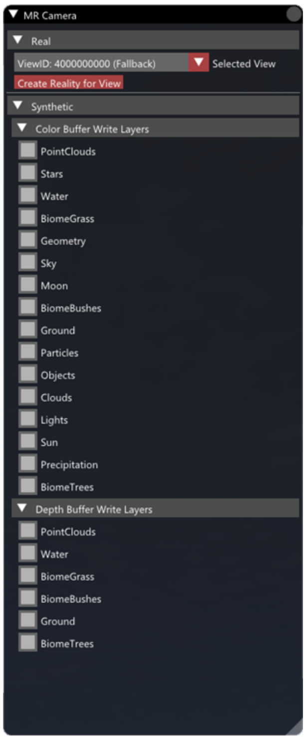
Panel Name	Description
Configuration Path	Input the model path to view. **<-(Left Alt + Left Arrow)** - Navigates to the previous class configuration view. **->(Left Alt + Right Arrow)** - Navigates to the next class configuration view. **^** - Goes one level up in the configuration path. **Page Up / Down** - Browses through the configuration-path history.
Ancestors Class	Displays the class hierarchy that the model inherits from.
Properties Class	Displays the values of all parameters that the model uses.

6.5.21 MR Camera

The MR Camera component window allows you to integrate sourced imagery from cameras, videos, or images and inject them into a rendered Mixed Reality / Virtual Reality scene.

The feature set of the MR Camera component is available at runtime using the Debug UI option.

> **NOTE**
>
> To make best use of the MR Camera feature set beyond runtime-only settings, and to be able to connect and configure imagery without using the Debug UI window requires the **MRCameraAspectAPI**, which is available in VBS IG SDK.



Follow these steps:

1. Press backquote (`) / tilde (~) to access the Debug UI (on page 376).
2. Select **MR Camera**.
3. Expand the settings and adjust the values, as required. For more information, see MR Camera Debug Settings (on the next page).

   > **NOTE**
   >
   > Any changes to the values only apply at runtime and do not persist after restarting the application.

6.5.21.1 MR Camera Debug Settings

Parameters Section	Description
Real	Options for injecting external source imagery into a view, either from a file or camera source.
Synthetic	Options for setting how the rendered layers get drawn, allowing the user to fine-tune integration of real-world elements into the IG by setting whether specific visual layers are drawn to the **Color Buffer** or **Depth Buffers**. For example, allowing occlusion of artificial objects behind correlated real-world terrain. • **Color Buffer Write Layers** - Select to enable any of the settings: ◦ PointClouds ◦ Stars ◦ Water ◦ BiomeGrass ◦ Geometry ◦ Sky ◦ Moon ◦ BiomeBushes ◦ Ground ◦ Particles ◦ Objects ◦ Clouds ◦ Lights ◦ Sun ◦ Precipitation ◦ BiomeTrees • **Depth Buffer Write Layers** - Select to enable any of the settings: ◦ PointClouds ◦ Water ◦ BiomeGrass ◦ BiomeBushes ◦ Ground ◦ BiomeTrees

7. Advanced Configuration

All VBS Blue IG configuration files are XML-based and can be directly edited to customize a specific setup.

These files are located in either of the following directories:

- Components folder:

  `\IG_Installation\Components\`

- Unified settings folder:

  `%LOCALAPPDATA%\Bohemia Interactive Simulations\VBS Blue IG\version\Settings\`

Explore the topics below to address your use case:

- Camera Locations (on the next page) - The camera can be set to jump by adding or delete locations.
- Headless Mode (on page 437) - Headless Mode allows VBS Blue IG to run without rendering the 3D scene to the screen.
- JRM Sensor (on page 438) - Allows vehicle models which utilize the custom JRM Sensor materials layering system to provide high-fidelity target signatures and sensor visualization in VBS Blue IG.

  > **NOTE**
  > This sensor requires a a valid license installed to the local system.

- Laser Configuration (on page 441) - Users can set the wavelength of lasers, such as the IZLID.
- Modify Mappings for CIGI (on page 443) - Users can add or override default mappings for CIGI.
- Settings Override (on page 449) - For scenarios where multiple IG instances share the same base settings but require specific unique settings to each IG.
- Startup Parameters (on page 454) - Use command-line parameters with VBS Blue IG to customize the application startup.
- VBS Geo Modifications - Manual Installation (on page 458) - When VBS Blue IG is connected to a VBS4 Host, any VBS Geo modifications from the current battlespace are automatically applied in VBS Blue IG.

7.1 Camera Locations

The camera can be set to jump by adding or delete locations.

Follow these steps:

1. Create a `CameraLocations.xml` file.
   a. Press backquote (`) / tilde (~) to access the Debug UI (on page 376).
   b. Select **Camera List Window**.
   c. Click **Save Current Camera Location**.
   d. Type a name in the **Location Name** text box.
   e. Click **Save Location**.

   The `CameraLocations.xml` file is now created.

   > **NOTE**
   > The file `CameraLocations.xml` is only created after a custom location has been saved in the **Camera List Window** from the Debug UI.

2. Open `CameraLocations.xml`, located in:

   `%LOCALAPPDATA%\Bohemia Interactive Simulations\VBS Blue IG\version\`.

   > **WARNING**
   > Make sure that VBS Blue IG is not running, when updating `CameraLocations.xml`.

3. Add / delete a camera location under `<CameraLocations>`:

```
<CameraLocations>
  <Location name="Bystrzyca Klodzka, Poland" default="true">
    <Position lat="50.3556124013" lon="16.6631134637" alt="485.3199998951" />
    <Orientation yaw="-126.71601619254956" pitch="16.125396110473574" />
    <Time year="2018" month="7" day="24" hour="8" minute="0" second="0" />
  </Location>
  <Location name="Bystrzyca Klodzka, Poland - Alternate">
    <Position lat="50.326774599" lon="16.577722068" alt="654.923768063" />
    <Orientation yaw="53.977448224912365" pitch="11.42970014535886" />
    <Time year="2018" month="7" day="24" hour="8" minute="55" second="0" />
  </Location>
   ...
</CameraLocations>
```

Parameter	Description
Location	Location specifications. The sub-parameters are: • `name` - Location name. • `default` - If set to `true`, then this is the default location that the camera is set to, when VBS Blue IG starts (default is `false`, so it can be left unspecified).
Position	Camera position. The sub-parameters are: • `lat` - Latitude. • `lon` - Longitude. • `alt` - Altitude.
Orientation	Camera orientation. The sub-parameters are: • `yaw` - Camera yaw. • `pitch` - Camera pitch.
Time	Camera scene time. The sub-parameters are: • `year` - Year (as YYYY). • `month` - Month (1-12). • `day` - Day (1-31). • `hour` - Hour (0-23). • `minute` - Minute (0-59). • `second` - Second (0-59).

4. Save `CameraLocations.xml`.

The camera location is now also added to / deleted from the Saved Camera Locations panel in the Camera List Window, in the VBS Blue IG UI. For more information, see Camera Location Selection (on page 200).

7.2 Headless Mode

Headless Mode allows VBS Blue IG to run without rendering the 3D scene to the screen. This mode is designed for CIGI Hosts that require many intersections to be done per-frame. The headless VBS Blue IG instance can be used to process all intersections, thus improving performance by allowing the other instances to focus on rendering the 3D scene.

7.2.1 Configuration

To properly configure Headless Mode, designate a single VBS Blue IG instance as the headless instance and every other instance as a regular client with intersection handling turned off.

Follow these steps:

1. Configure the headless Instance:
   - Add the command line argument `-headless`, when launching BlueIG.exe.

     VBS Blue IG launches into a black screen without 3D rendering.
2. Configure all other instances:
   a. Open the CigiProtocol settings in the following file:

      `%LOCALAPPDATA%\Bohemia Interactive Simulations\VBS Blue IG\<version>\Settings\CIGI.xml`.

   b. Modify the `Queries` setting to `<Queries Enabled="false">`.

> **NOTE**
> In Headless Mode, you are still able to interact with the Debug UI menu and the F1 Rendering Options.

7.3 JRM Sensor

The JRM Sensor thermal imaging optics can be accessed within VBS Blue IG via the **IG View Manager**. Vehicle models which utilize the custom JRM Sensor materials layering system provide high-fidelity target signatures and sensor visualization in VBS Blue IG.

Configure the the JRM sensor with the following:

- Debug UI Set Up (below)
- CIGI Setup (on page 440)

FEATURE NOTICE

To activate the JRM Sensor in **VBS Blue IG** requires a valid license with the JRM SDK installed to the local system. If the JRM SDK is installed but a valid license is not detected, the JRM Sensor will not activate when selected.

To gain full access to the JRM Sensor, please contact sales@bisimulations.com for further information.

7.3.1 Debug UI Set Up

Follow these steps to activate the JRM Sensor:

1. Launch VBS Blue IG.
2. Press backquote **`** / tilde **~** to open the Debug UI.
3. Press **Esc** to show the mouse cursor.
4. Select **View Manager**.
5. Expand the **Fallback View > Sensor** dropdown menus.
6. Expand the **Type** dropdown menu and select **JrmSensor**.

   NOTE

   Upon the initial launch of the JRM Sensor, a wait time of 12 to 15 minutes is expected while the JRM SDK builds the sensor cache. During this period, a blank screen will be displayed within VBS Blue IG. Once the sensor cache has been built, subsequent activations of the JRM Sensor on the same system will not have this delay.

7. While the JRM Sensor is active, expand the **Parameters** dropdown menu.
8. Expand any of the available parameters dropdown menus to access and modify them.

9. To commit parameter modifications, expand the **Apply Parameters** dropdown then select the **Apply** checkbox button.

> **WARNING**
>
> Any modifications made to the JRM Sensor Environment Parameters (other than Time of Day) require that the sensor cache be rebuilt by the JRM SDK to accommodate for adjusted atmospherics. This process may take 12 to 15 minutes to complete. During this time, any further modifications made to other JRM Sensor parameters will not be applied or displayed within VBS Blue IG until the JRM SDK cache rebuild process has completed.

Image-1: JRM Sensor options in Debug UI.

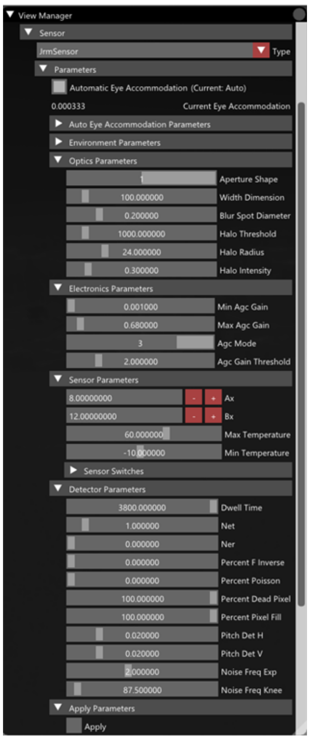


Image-2: JRM Sensor enabled in scenario.

7.3.2 CIGI Setup

CIGI mapping for the JRM Sensor is not included by default with the VBS Blue IG installation. To map this sensor, open the `ViewTypes.mapping` file in the following folder:

`\IG_Installation\data\BlueProduct\mappings\`

Add this line:

```
cigi-view-type:3 > sensor:JrmSensor, version:1
```


7.4 Laser Configuration

Users can set the wavelength of lasers, such as the IZLID. This enables users to change the color of lasers and set visibility based on sensor modes.

In order for a laser to be visible in visible light, the wavelength must be set between 380-780nm. The color of the laser depends on the wavelength. For a laser to only be visible in NVGs, the wavelength must be set to 781nm or greater.

VBS Host

When using VBS as a host, the wavelength of lasers can be changed.

Follow these steps:

1. Open the following file:

   `%LOCALAPPDATA%\Bohemia Interactive Simulations\`
   `VBS Blue IG\<version>\Settings\VBSExternalNetworking.xml`

2. Modify the `Lasers>DefaultWavelength` value, as required.
3. Modify the `Lasers>LaserAnimationDistance` value, as required.

> **NOTE**
> This change is applied to all lasers.

CIGI Host

For customers using a CIGI host, the wavelength of an individual laser can be set when the laser is created using the **Laser Control** CIGI packet (see the online ICD Reference (https://manuals.bisimulations.com/cigiicd/icd_23_2/index.html)) by passing the desired wavelength as a parameter to the Wavelength field. Because this packet is specific to one laser, the user can set different wavelengths for each laser in their scene.

Laser Collision Test

Use the `UseLaserEndPointFromVBS` setting, located in the `VBSExternalNetworking.xml` file to determine whether collision checks for lasers are performed on the VBS host or VBS Blue IG client.

If `UseLaserEndPointFromVBS` is set to true, then the collision checks for all lasers are done on the host (VBS) side and passed along to VBS Blue IG to render. This setting is set to true by default. Visually, this means that the collision (or end) point of the laser will always match between VBS and VBS Blue IG. However, it is possible that the laser will appear to be going through objects or stopping too soon on VBS Blue IG if the object it is hitting on VBS is not properly correlated with VBS Blue IG.

If `UseLaserEndPointFromVBS` is set to false, then the collision checks for all lasers occur on the VBS Blue IG side. Visually, this means that the endpoint of the laser on VBS Blue IG may not match the endpoint on the VBS host. However, since the collision check is done on the VBS Blue IG side, the endpoint will never appear to go through objects, or stop too soon.

7.5 Modify Mappings for CIGI

CIGI mappings are stored in the following directory:

`\IG_Installation\data\BlueProduct\mappings\Default\`

The default mappings are contained within the `\Default\` subfolder, and users can add or override these mappings by placing additional mapping files in additional subfolders.

These mappings load after the default mappings, and any mappings with identical mapping identifiers override the default mappings.

```
input_qualifier:input_instance[, input_qualifier:input_instance] > output_
qualifier:output_instance[, output_qualifier:output_instance]
```

The mapping output qualifiers can be in the format of VBS class name or model `.p3d` path as shown in the following:

- `cigi-entity-type:1 > vbs:vbs2_us_mc_rifleman_w_m16a4`
- `cigi-entity-type:2 > blue-model:vbsblue\debug_primitives\arrow_end_5m_black.p3d`

To create invisible parent objects, mappings can be made available but not visible using the following identifier:

EXAMPLE

```
cigi-entity-type:0 > invisible:1
```

Additional types of mapping input identifiers can be used throughout the mapping system.

For more information, refer to the VBS Blue IG ICD documentation for your version of CIGI, and the header comment description in the default mapping files to learn which identifiers are used for which mapping types.

7.5.1 Adding Mappings

Add new mappings to the appropriate mapping file.

Follow these steps:

1. Open the mapping file that you want to modify. For example:

   `\IG_Installation\data\BlueProduct\mappings\Default\Materials.mapping`

2. Choose a unique CIGI ID not exceeding 65535. Add a line in the mappings file with this format:

   `cigi-entity-type:## > vbs:vbs_my_custom_model_class`

   Then ensure that the host uses this number for the entity type in the entity packet.

7.5.2 Adding Custom Content to VBS Blue IG

After a model has been packed, add it to VBS Blue IG.

Follow these steps:

1. Add to the packed content to the `\content\` directory in your build:

   `\IG_Installation\myData\Blue\content\MyCompany\MyContent.pbo`

2. Add a new mapping as described in Adding Mappings (on the previous page).

7.6 Recording and Playback of VBS External Networking traffic

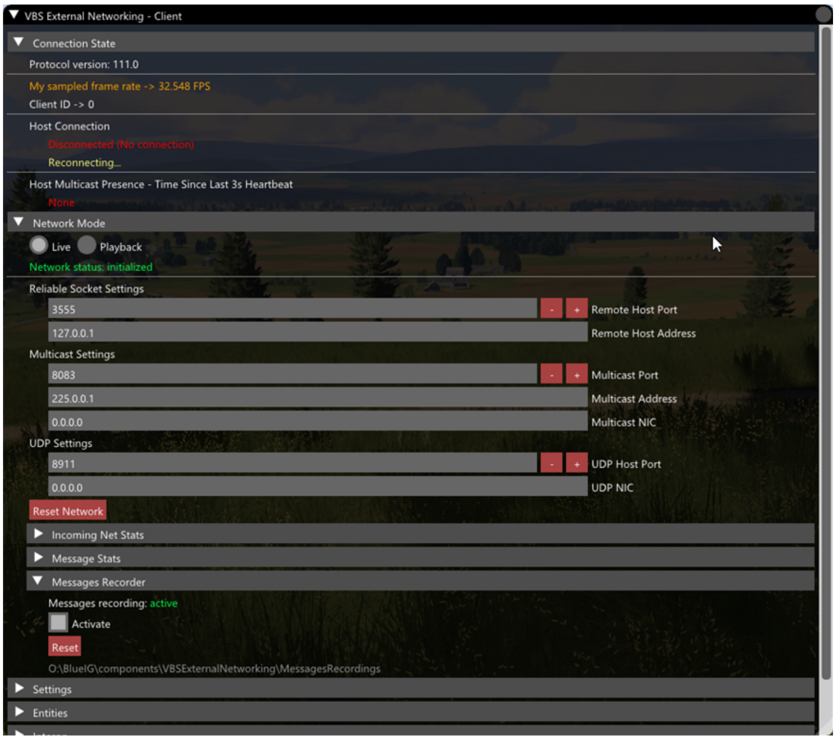
Network traffic received by VBS Blue IG from a VBS4 host can be recorded into a file and played back with an alternative **playback** network handler. The alternative network handler simulates the incoming network traffic by reading the messages from the recording file. The application responds to the simulated traffic in the same way as it does to real live network traffic.

Switching VBS Blue IG to the playback mode is done by using the Debug UI (on page 376). It is also possible to record the incoming network traffic to a file during the normal (live) network operation.

- Traffic Recording (below)
- Traffic Playback (on page 447)

7.6.1 Traffic Recording

Enable recording of the incoming network traffic in VBS Blue IG.



Follow these steps:

1. Press backquote (`) / tilde (~) to access the Debug UI (on page 376).
2. Select **VBSExternalNetworking (IG)**.
3. Click the drop-down arrow for **Network Mode**.

4. Select **Live** mode.

> **NOTE**
>
> This is the original Blue IG network mode.

5. Expand the **Messages Recorder** section.
6. **Optional**: Select the target folder.
7. Click the **Activate** checkbox.

All incoming traffic, connection and disconnection events, are recorded into the file `IncomingTraffic-{date-Time}.mtr`. The file is saved automatically, after deactivating the message recording or exiting VBS Blue IG.

> **WARNING**
>
> **Integration with VBS4**: In order to properly capture a VBS4 mission load event, recording must be active before creating a VBS4 lobby or VBS4 dedicated server instance. If the mission load event is created before recording starts, other captured messages will still be replayed, but VBS Blue IG will not display the desired scene.

7.6.2 Traffic Playback

Switch VBS Blue IG to a playback mode.

Follow these steps:

1. Press backquote (`) / tilde (~) to access the Debug UI (on page 376).
2. Select **VBSExternalNetworking (IG)**.
3. Click the drop-down arrow for **Network Mode**.
4. Select **Playback** mode.
5. In the text field, paste the full path of the recording file to play, then click **Open**.
6. To restart the playback as needed (for example, when the end of the recording is reached), click **Restart**.
7. To see details of the open recording file, expand the **Current Recording** section.

   This file contains the total count of messages and connection / disconnection events, as well as the total duration of the recording.

8. To play the recording, expand the **Playback** section, and select from the following options:
   - **Auto Play** - Click this option to activate or deactivate **automatic playback** at any moment. The playback is based on the timestamps and time of the messages that passed since the beginning of the recording.
   - **Step Over** - Click this option to initiate a **manual step over playback** where each click progresses the playback to the next message to be processed, regardless of its timestamp.

   The section shows the last processed message, including index and timestamp (**Last message / event**), and the next processed message **Next message / event**.
9. Convert the recording file to a human readable `JSON` (with messages data details) or `CSV` (without data details) files, as needed:
   a. Input the output file path.
   b. Click **Export to JSON** or **Export to CSV**.

   The section shows progress of the playback using 2 ratios:
   - **Messages Count** - Processed messages count to total messages count.
   - **Duration** - Processed time to total duration.
10. To play the recording, expand the **Playback** section.

> **NOTE**
>
> If no entities / events are visible but messages are being processed, see VBS External Networking traffic recording / playback (on page 465).

> **NOTE**
>
> Geo modifications do not work during playback unless the original battlespace is still running, because the server that provides the geo package may not be running anymore.

7.7 Settings Override

It may be necessary to have multiple IG instances share the same base settings (such as video settings), but have specific unique settings to each IG (such as IP addresses).

This can be achieved by using settings override. The settings override feature applies an additional layer on top of the base settings, overriding any of the settings specified.

Settings override follows the same XML structure of settings. However, only specific settings that need to be overridden should be specified.

> **NOTE**
>
> Settings that are overridden cannot be saved in the Settings GUI, and will show an icon in the GUI.

An example of such a configuration may be 3 IG instances, sharing the same set of base settings, with each IG instance having their own unique set of settings overrides.

Follow these steps:

1. Start a single VBS Blue IG instance with the required `-productDir`. Typically when configuring multiple IGs, a Windows network share is utilized. However, the directories can be local to each PC, as well:

   `BlueIG.exe -productDir="\\network.share\BlueIG\SharedProductDir"`

2. Configure the shared settings as required.
3. Copy the `\SharedProductDir\Settings\` directory to the locations of the override directories, for example, `IG1Settings`, `IG2Settings`, `IG3Settings`.

4. Identify which settings need to be overridden. For example, IG1 may need to send CIGI packets to a specific IP address. IG2 may need to enable video streaming on startup. IG3 may need a different window size. Therefore, in this example, delete all `.xml` files, and all XML nodes and attributes that do not contain settings that need to be overridden.

a. The `\IG1Settings\` directory just contains a `CIGI.xml` file, with:

```
<?xml version="1.0"?>
<CIGI>
    <Sessions>
        <Session>
            <Network>
                <Send Address="192.168.1.50" />
            </Network>
        </Session>
    </Sessions>
</CIGI>
```

b. The `\IG2Settings\` directory just contains the `Streaming.xml` file with the following:

```
<?xml version="1.0"?>
<Streaming>
    <StreamOnStartup>true</StreamOnStartup>
</Streaming>
```

c. The `\IG3Settings\` directory just contains the `VideoSettings.xml` file with the following:

```
<?xml version="1.0"?>
<VideoSettings>
    <Window>
        <window_size_x>1280</window_size_x>
        <window_size_y>720</window_size_y>
    </Window>
</VideoSettings>
```

5. Start VBS Blue IG with both `-productDir` and `-settingsOverrideDir` command line arguments.

```
BlueIG.exe -productDir="\\network.share\BlueIG\SharedProductDir"
 -settingsOverrideDir="\\network.share\BlueIG\IG1Settings"
```

```
BlueIG.exe -productDir="\\network.share\BlueIG\SharedProductDir"
 -settingsOverrideDir="\\network.share\BlueIG\IG2Settings"
```

```
BlueIG.exe -productDir="\\network.share\BlueIG\SharedProductDir"
 -settingsOverrideDir="\\network.share\BlueIG\IG3Settings"
```

6. Each IG shares the same product directory and base set of settings. They also use their unique set of overridden settings as specified.

7.7.1 Overriding Branch Arrays

While regular values are completely replaced when overridden, overriding branch arrays is more complex since it is often preferred to modify only a certain sub-value of the element branch instead of completely replacing it.

To enable this, the override behavior can be specified using a special `:override` attribute applied to the overriding element. The following behaviors are available:

- **Merge**
  - Merge the existing element with the override one. Any provided override values overwrites existing ones. Remaining values remain unmodified.
  - If the target element does not exist, it is appended.
  - If no behavior is specified, this is the default.
- **Replace**
  - Discard the existing element and replace it with the override one. Non-provided values are set to defaults as defined in the branch array template.
  - If the target element does not exist, it is appended.
- **Delete**
  - Delete the existing element and do not replace it with anything.
  - If the target element does not exist, nothing is changed.

NOTE

The parent branch array is not considered modified unless an element is deleted, replaced or appended. Merging overriding values into an existing element only sets those values as overridden, not the parent branch array itself.

Examples:

- Override the first CIGI session to use CIGI version 3.3:

```
<CIGI>
  <Sessions>
    <Session>
      <CIGI Version="3.3" />
    </Session>
  </Sessions>
</CIGI>
```

- As above, but with the merge behavior explicitly specified:

```
<CIGI>
  <Sessions>
    <Session :override="merge">
      <CIGI Version="3.3" />
    </Session>
  </Sessions>
</CIGI>
```

- Append a second session running on a different port, but using defaults for the rest:

```
<CIGI>
  <Sessions>
    <Session /> <!-- Empty merge = modify nothing in the first session -->
    <Session> <!-- Merge into nothing = append -->
      <Network>
        <Receive Port="8005" />
        <Send Port="8006" />
     </Network>
    </Session>
  </Sessions>
</CIGI>
```


- Delete the second session:

```
<CIGI>
  <Sessions>
    <Session /> <!-- Empty merge = modify nothing in the first session -->
    <Session :override="delete" />
  </Sessions>
</CIGI>
```

- Replace the first sessions with the defaults:

```
<CIGI>
  <Sessions>
    <Session :override="replace" />
  </Sessions>
</CIGI>
```

7.8 Startup Parameters

Use command-line parameters with VBS Blue IG to customize the application startup.

Follow these steps:

1. Open Windows CMD.
2. Run VBS Blue IG using any of the parameters in the table below.

   Usage: `\IG_Installation\BlueIG.exe -scalable`

Parameter	Description	Default Value	Example Usage
`-dataAdapter`	Enables a second GPU to process terrain and data. This enables parallel processing of rendering and data refinement, improving performance and reducing FPS stutter. The number used is the index of the connected device, where **0** is always the primary adapter, and is typically **1** when the system has two graphics cards available.	<not set>	`-dataAdapter`
`-defaultViewConfig`	View Configuration - The path to the XML file containing configuration settings for View definitions. This may be a file on the local PC or on a network share.	**File location**: `\IG_Installation\components\BlueIGViewSystems\Config\DefaultViewConfig.xml`	`-defaultViewConfig="C:\My Settings\DefaultViewConfig.xml"` `-defaultViewConfig="\\network.share\Blue IG\DefaultViewConfig.xml"`
`-domeprojection`	Warp / Blend - Enables DomeProjection Warp and Blend. **NOTE** This parameter cannot be used together with `-scalable` or `-warp`. For more information, see DomeProjection Warping Setup (on page 137).	<not set>	`-domeprojection`

Parameter	Description	Default Value	Example Usage
`-fullDump`	Automatically configure crash dumps to dump full memory information, which provides more information for debugging than the default mini crash dumps.	<not set>	`-fullDump`
`-geoPackage`	Designates a Geo Project file `.geo` to be used at startup. Parameter points to a specific URL and streams it to the `data\BlueProduct\earth\Geo` directory. **Example:** `-geoPackage=http://xxx.xxx.xxx.xxx:25500/xxx.geo`	<not set>	`-geoPackage`
`-headless`	Allows VBS Blue IG to run without rendering the 3D scene to the screen. This mode is designed for CIGI Hosts that require many intersections to be done per-frame. For more information, see Headless Mode (on page 437).	<not set>	`-headless`
`-hmd`	Enables usage of a connected Head Mounted Display (HMD), such as an Oculus Rift or an HTC Vive. If the HMD is not connected, or not supported, an error is displayed on start up, and the application does not run. For more information, see Command Line Launch Options for HMDs (on page 68).	<not set>	`-hmd`
`-interopHost`	Deprecated. Equivalent to `-vbsHostNet`.	<not set>	`-interopHost`
`-interopIP`	Deprecated. Equivalent to `-vbsHostIP`.	127.0.0.1	`-interopIP=192.168.1.3`

Parameter	Description	Default Value	Example Usage
`-master`	IG Multichannel Synchronization - Defines this instance as the master instance to synchronize all clients from. See Views Settings (on page 361) for more information on master clients.	<not set>	`-master`
`-noSplash`	Disables the loading and unloading of splash screen.	<not set>	`-noSplash`
`-productDir`	The path to the directory which contains the settings XMLs, log files, crash dumps, and other generated data. This may be a directory on the local PC or on a network share.	`%LOCALAPPDATA%\Bohemia Interactive Simulations\VBS Blue IG\`*`version`*`\`	`-productDir="\\network.share\Blue IG\ProductDir"`
`-scalable`	Warp / Blend - Enables Scalable Warp and Blend. **NOTE** This parameter cannot be used together with `-domeprojection` or `-warp`. For more information, see Scalable Warping (on page 427).	<not set>	`-scalable`
`-settingsOverrideDir`	The path to the directory which contains settings override configuration. For more information, see Settings Override (on page 449). This may be a directory on the local PC or on a network share.	<not set>	`-settingsOverrideDir="C:\My Settings\Overrides"` `-settingsOverrideDir="\\network.share\Blue IG\Overrides"`
`-vbsHostIP`	Replaces legacy `interopIP=address` argument. Allows specifying the IP address of the machine running VBS4 with `-vbsHostNet` (or `-interopHost`) defined.	127.0.0.1	`-vbsHostIP=10.3.50.44`

Parameter	Description	Default Value	Example Usage

`-vbsHostNet`	VBS Host - Used to enable / disable **VBSExternalNetworking** host component on VBS4. Overrides **Networking** > **EnableVBSHostNetworking** setting.
	VBS Blue IG - Used to enable / disable **VBSExternalNetworking** component on VBS Blue IG. Overrides **Networking** > **EnableVBSHostNetworking** setting.
		VBS Host:
		0
		VBS Blue IG:
		1
			`-vbsHostNet`
			`-vbsHostNet=1`
			`-vbsHostNet=0`

`-warp`	Warp / Blend - Enables Generic Warping. **NOTE** This parameter cannot be used together with `-domeprojection` or `-scalable`. For more information, see Generic Warping (on page 429).	<not set>	`-warp`

7.9 VBS Geo Modifications - Manual Installation

When VBS Blue IG is connected to a VBS4 Host, any VBS Geo modifications from the current battlespace are automatically applied in VBS Blue IG.

However, it is also possible to manually install modifications created in VBS4 to VBS Blue IG.

Follow these steps:

1. Navigate to the directory with the VBS4 Battlespace containing the VBS Geo modifications to install.
2. Copy `\Geo\geoproject.geo` to:

   `\IG Installation\data\BlueProduct\earth\Geo\`.

   > **NOTE**
   > Although the file may be renamed to allow multiple modifications to be loaded at once, the `.geo` file extension must be retained.

   > **WARNING**
   > Sub-directories are not supported.

   > **NOTE**
   > It is unnecessary to restart VBS Blue IG since any modifications will be applied in real-time.

8. Troubleshooting

This section covers how to fix common issues that can occur when setting up or using VBS Blue IG including the following:

- Cannot Connect to VBS Host (below)
- Cannot Connect to CIGI Host (on the next page)
- Verify Build Integrity (on page 461)
- Entities show up from VBS but do not Update (on page 462)
- Unmapped Entities (on page 462)
- DIS entities from external simulators not appearing (on page 463)
- VBS Blue IG Suffers Performance and Stutter Issues (on page 463)
- Excessive Fog Accumulates over Time (on page 464)
- Create Profile Captures (on page 464)
- VBS External Networking traffic recording / playback (on page 465)

8.1 Cannot Connect to VBS Host

Verify Network Connection:

Verify that the computers are able to see each other on the network.

- On the host computer, open Windows CMD and run the following command: `ping <client IP address>`
- On the client computers, open Windows CMD and run the following command: `ping <host IP address>`

Verify Network Settings:

Verify the Host address, multicast address, and Ports are matching by checking the `VBSExternalNetworking.xml` file for both the host and client computers.

- For VBS Blue IG:

  `\%LOCALAPPDATA%\Bohemia Interactive Simulations\VBS Blue IG\version\Settings\`

- For VBS4:

  `\%LOCALAPPDATA%\VBS4\Settings\`

Verify IG Connection Status:

See if the IG shows as connected in the Debug UI VBSExternalNetworking Section.

1. On the IG client computer press **Escape**, then press the backquote **`** / tilde **~** key to bring up the Debug UI.
2. Click **VBS External Networking**.
3. Click the **Networking** drop-down menu.
4. Verify that **Host TCP Connection** says Connected and that the host is listed in **Host Multicast Presence**.

> **NOTE**
> A scenario (or an AAR session) must be running on the VBS host before a connection can be successfully established by VBS Blue IG.

Verify Ports:

- It is possible that another application or another plugin is utilizing the same port causing a conflict with VBS4 to VBS Blue IG communication. Changing ports will verify this.
- netstat - Running this command in a Windows command window will show all active connections. This allows you to see if a port is already being used on your computer.

Verify TTL:

VBS Blue IG only accepts multicast datagrams with the TTL value of 1, whereas the default value in VBS4 is 16. If the Host Multicast Presence can not be established after resolving any previously mentioned issues, consider setting the Multicast TTL setting in `VBSExternalNetworking.xml` in VBS4 to a value of 1.

8.2 Cannot Connect to CIGI Host

Verify Network Connection:

Verify the computers are able to see each other on the network.

- On the host computer, open Windows CMD and run the following command: `ping <client IP address>`
- On the client computers, open Windows CMD and run the following command: `ping <host IP address>`

Verify Network Settings:

Verify the Host address, IG Update address, and Ports are set correctly in the startup batch. Verify the Host's settings for sending data to the IGs matches the IGs settings such as Address and ports for sending data to the IGs. If using a broadcast or multicast address try unicasting to a single computer.

Verify Ports:

- It is possible that another application or another plugin is utilizing the same port causing a conflict with host to VBS Blue IG communication. Changing ports will verify this.
- netstat - Running this command in a Windows command window will show all active connections. This allows you to see if a port is already being used on your computer.

Verify Packets on the Network:

Using an application called Wireshark you can verify that CIGI packets (in the filter type **cigi**) are reaching the multicast address as well the port. You will need this on both computers to make sure that both the host and the IG computer network interfaces are seeing the CIGI traffic on the network. If you are not seeing the network traffic this could indicate a problem with the router or network device being used.

8.3 Verify Build Integrity

To ensure the downloaded and installed build matches the expected set of files without any corruption or modifications, the verify the build for integrity.

Follow these steps:

1. Run `VerifyBuildIntegrity.exe` at:

   `\IG_Installation\tools\support\`

   The installation directory is scanned and compared to the known set of build checksums.

   > **NOTE**
   > This may take several minutes.

2. A summary is presented with a list of valid, missing, extra, or modified files in the build.
   - If VBS Blue IG has been run, some differences may be expected, such as cache and log files. Any custom content will also show as extra files.
   - If any missing or modified files are present, the build should be re-installed.
3. A report in `.csv` format is saved to:

   `\IG_Installation\tools\support\`

4. If the results contain any unexpected changes, please uninstall then re-install VBS Blue IG.
5. To check for a corrupt download, the **Verify Checksum** option in the installer should be enabled. The installer files may need to be re-downloaded if they are corrupted or missing.

8.4 Entities show up from VBS but do not Update

This is usually an indication that you have set the right Host address but not the correct Multicast settings. See “Verify IG Connection Status” under “Can’t Connect to VBS Host” to see if TCP is connected and host is present in the **Host Multicast Presence**.

If Multicast does not show the host then try using a different multicast address or port. If the problem persists, this may indicate that your router / access point / network is causing an issue when attempting to use multicast to the IG Machines. Please check with your IT Administrator about the router being used since it needs to support **IGMP Multicasting**.

8.5 Unmapped Entities

CIGI as Host:

Verify that the entity is showing as unmapped within the editor. This can be done through the Debug UI. There is a list of all entities the IG has received (for both and CIGI and VBS Host).

For CIGI you need to confirm the class name of the entity you want to spawn. Use VBS if you are trying to find the class name of a vehicle that is supported by VBS Blue IG. When you spawn an object within the editor of VBS you can see its class name when selecting the object in the bottom bar if you have Object Class Name enabled for the help bar.

If this is a custom model then you will need the class name as it was defined in the config.

VBS as Host:

1. Access the Debug UI by pressing backquote **`** / tilde **~**.
2. Open the **Entity List Window** in order to see all entities.
3. Determine in the open window whether the **Mapping** or **Model** field says <unmapped>.
4. After verifying that the entity shows up either provide a mapping or include the content.
5. If the entity does not appear, then it is possible that this entity is blacklisted because of issues. Please contact support for more information and assistance.

8.6 DIS entities from external simulators not appearing

In order for DIS entities to appear in your VBS host scenario, the **interopForwarding** option must be set within VBS4.

For more information, see Enabling DIS Entities (on page 46).

8.7 VBS Blue IG Suffers Performance and Stutter Issues

The following options can be used to improve VBS Blue IG performance and reduce stutters.

Disable CPU Underclocking:

1. Select the Windows key on the keyboard.
2. Type *Power & Sleep settings* and click on **Power & Sleep settings**.
3. Click **Additional Power Settings**.
4. Expand **Show Additional Plans**.
5. Select **High Performance**.

Disable GPU Underclocking:

1. Select the Windows key on the keyboard.
2. Type NVIDIA Control Panel and select it.
3. If opening for the first time, click **Agree** and **Continue**.
4. Expand **3D Settings**.
5. Click **Manage 3D Settings**.
6. Change the Power Management Mode to **Prefer Maximum Performance.**

8.8 Excessive Fog Accumulates over Time

To prevent fog from VBS hosted missions accumulating, set the base and ceiling values in VBS.

1. In the VBS Editor, select **Tools > Scenario Settings**.
2. Under **-Weather-** apply the minimum values to the following settings:
   - Set **Fog Base / Ceiling** to **0** and **50**.
   - Set **Wanted Fog Base / Ceiling** to **0** and **50**.
3. Click **OK** and save the mission.

8.9 Create Profile Captures

Profiling can be used to determine the cause of poor performance, and can capture frame timing and highlights where time is being spent in each frame. VBS Blue IG uses a tool called **microprofile** for this functionality. Users of VBS Blue IG are not expected to be able to analyze the frame timing, but it is useful to provide to Bohemia Interactive Simulations when experiencing poor performance.

Follow these steps:

1. Open Metrics (on page 404) and select **Enable Profiling**.

   > **NOTE**
   > Profiling has a performance impact and should not be enabled during regular usage.

2. The default web browser will open to the microprofile page.
3. Enable **Control / All**.

   A frame time graph should appear at the top of the webpage.
4. Reproduce the poor performance or stutter.

   The frame time graph will be taller for frames with worse performance (taking more time to render the frame).
5. **Optional:** Press **Spacebar** to pause VBS Blue IG. This action captures an exact section of the frames. Press **Spacebar** again when ready to un-pause.
6. **Right-click and drag** a region on the frame time graph that shows the poor performance, then press **Enter**.

   The selected region should not be too large; less than 10 frames is typically an appropriate region to select.

   A new page will open with the capture.

7. Save the page (**Ctrl+S**) as an `.html` file. This file can then be sent to Bohemia Interactive Simulations for further analysis.
8. Multiple captures can be created.

   Each capture will be its own `.html` webpage.
9. Restart VBS Blue IG.

8.10 VBS External Networking traffic recording / playback

When playing back a recording, if no entities / events are visible but messages are being processed, the mission load event may not have been properly captured. Confirm that the recording is activated as part of the workflow by following the steps below:

1. Before creating a VBS4 lobby, activate the recording.
2. Before launching a dedicated server VBS4 instance, activate the recording.