HOLOLIGHT

HOLOLIGHT
STREAM

Hololight Stream SDK for Unreal Engine

Holo-Industrie 4.0 Software GmbH | Carl-Zeiss-Ring 19 | 85737 Ismaning

Table of Contents

1. Requirements

- Unreal Engine
- Hololight Stream Unreal Plugin 2025.0

1.1. Supported versions of Unreal Engine

- Unreal Engine 5.3
- Unreal Engine 5.4
- Unreal Engine 5.5

NOTE	Each version of Unreal Engine requires the corresponding version of the Hololight Stream project plugin.

1.2. Supported client devices

Hololight Stream supports the following client devices:

- Apple iOS and iPadOS
  Connect, but touch input unsupported
- Apple Vision Pro
- HTC VIVE Focus 3
- HTC VIVE Focus Vision
- HTC VIVE XR Elite
- Lenovo ThinkReality VRX
- Magic Leap 2
- Meta Quest 2
- Meta Quest 3
- Meta Quest 3s
- Meta Quest Pro
- Microsoft HoloLens 2
- PICO 4 Ultra
- Web browser - Chromium
- Web Browser - Safari
- Windows

1.3. System requirements

- NVIDIA Graphics Card
  Hololight Stream supports many of the leading video compression codecs. However, whether a codec is available to use depends on two factors:
  - Does the device running Hololight Stream Client support the codec?
  - Does the GPU of the server machine running the application support the codec?

  To figure out which codecs your machine running the server application supports, you should check your GPU maker's official documentation. Remember, Hololight Stream only officially supports NVIDIA GPUs, though others may work.

  For information on what NVIDIA GPUs support which codecs, check out NVIDIA's **Video Encode and Decode GPU Support Matrix**.

NOTE	How good an application's performance is depends on a variety of factors. Applications using complicated models will require more advanced hardware regardless if the client device and server machine share compatibility with a particular codec.

  Refer to the Hololight Stream Client documentation for details.

- DirectX 11 or DirectX 12
- Windows 10 or Windows 11
- Stable network connection with enough bandwidth

2. Set up Hololight Stream for Unreal Engine

2.1. Install Unreal Engine

You can install Unreal Engine in two different ways:

1. **Download and install Unreal Engine using the Epic Games launcher.**
2. **Build Unreal Engine from source.**

NOTE	Hololight Stream is compatible with Unreal Engine source code builds, but only if there are no major differences in the code that Hololight Stream depends on. If you change any source code Hololight Stream depends on it won't work.

2.2. Install the Hololight Stream plugin

1. Using File Explorer in Windows, navigate to the folder for the Unreal Engine project you just created.
2. Create a folder named "Plugins" if there isn't one already.
3. Place the Hololight Stream plugin package in the folder. The "Plugins" folder needs to be in the same location as the Unreal project file (which has the extension ".uproject") for your application.

NOTE	Normally, the Unreal Editor automatically detects a new plugin like Hololight Stream. If it doesn't, select **Edit > Plugins > Installed**. Find **Hololight Stream**. Make sure there is a checkmark next to it that enables Hololight Stream.

2.3. Configure Unreal Engine project for XR

Unreal Engine supports XR head-mounted displays like Meta Quest devices and Microsoft HoloLens 2. This means any Level will work with these and other similar devices when using this mode.

NOTE	In addition to what running the application using VR does to the camera, you can implement your own logic too. However, if your logic affecting the camera interferes with VR, you will see unexpected behavior. This includes elements like camera animations and screen-based UI.

In addition to just using the camera, you can also create a special kind of Pawn called a VRPawn. You can then add a camera, controllers, and other necessary components to the VRPawn.

Once you set everything up, Select **Advance Play Settings > VR Preview** to set the Play mode to **VR Preview**.

2.3.1. Using the VR Template

The VR Template contains all the necessary Blueprints, input mappings, and more to create a basic VR game prototype. To create a new project in Unreal Engine using the VR Template:

Set up using default Unreal Engine VR Template

1. Launch Unreal Engine.
2. Select **Games > Virtual Reality**.
3. Under **Project Name**, name your project.
4. Select **Create**.

NOTE	You can add the VR Template configuration to an existing project. Go to **+ Add > Add Feature or Content Pack > Virtual Reality**.

Disable Open XR plugins

The Unreal Editor VR Template is designed to work with Unreal's OpenXR plugin. Hololight Stream does work with Unreal Engine the same way OpenXR does. You need to make a few changes to the VR Template to work with Hololight Stream.

If you use the Unreal Engine's VR Template, it uses OpenXR plugins by default. OpenXR can conflict with Hololight Stream, so you need to disable all OpenXR and OpenXR-related plugins.

1. Select **Edit > Plugins**.
2. Select **ALL PLUGINS**.
3. Go through the list and disable all OpenXR-related plugins. This includes:

   Open XR-related plugins
   - OpenXR
   - OpenXR visionOS
   - OpenXREyetracker
   - OpenXRHandTracking
   - OpenXRMsftHandInteraction
   - OpenXRViveTracker
   - XRScribe

NOTE	In addition to OpenXR plugins, you need to disable all other XR-related plugins that may be in the project. This includes, but is not limited to, Meta XR and PICO XR plugins.

4. Close and then restart Unreal Editor.

Use correct motion controller motion sources

VR Template has controller motion sources on Grip motion sources. Hololight Stream does not support Grip motion sources, but supports Left/Right, Palm and Aim motion sources. On the VRPawn, there are Motion Controller components for getting motion sources LeftGrip and RightGrip. On these components, Motion Source should be changed to Left/Right. With this change, the positioning of the visualization model also needs to be adapted to fit the position of the motion sources. Also, on the GrabComponent Blueprint, inside the GetHeldByHand function, there is a check of controller motion source with the name LeftGrip. This name also needs to be changed to Left.

Enable passthrough

To enable passthrough in Unreal Engine:

1. Select **Edit > Project Settings**,
2. Set **Alpha Output** to true.

To use passthrough on the Level in the VR Template, remove or hide the Sky Sphere.

NOTE	In Unreal Engine 5.4 and earlier Alpha Output is Enable alpha channel support in post processing. Set this to Allow through Tonemapper.

Configure input mappings

VR Template does not have all the input mappings needed to support all Hololight Stream enabled devices. Hololight Stream registers the devices to the Unreal Engine. After that you can add devices to the Input Mappings or you can use the example Hololight Stream Input Mappings instead.

Change **Project Settings > Enhanced Input > Default Mapping Contexts** to the ones provided with the plugin.

Set up using Hololight Stream Unreal Engine VR Template

Hololight Stream comes with its own ready-to-go modified version of Unreal's VR Template. To find the template select **Content Browser/Drawer > Settings > Show Plugin Content**.

NOTE	The Hololight Stream VR Template uses Unreal Engine 5.5 and is not available in older versions of Unreal Engine.

Changes in the Hololight Stream Unreal Engine VR Template

Motion Controller Sources

To track motion controllers and hands, Motion Controller Components use Motion Sources. In the Unreal Engine VR Template, the base sources are LeftGrip and RightGrip. Instead Hololight Stream supports Left and Right as the base sources. Because of this, the VRPawn and GrabComponent are modified with the correct motion sources.

Controller and hand visualization

The package includes models for the various controllers for the different supported clients. When the application runs and a client connects to it, the application uses these models to visualize the user's controllers. The application will also visualize tracked hands. VRPawn visualizes the controllers or hands when a client connects.

Custom game instance

A custom GameInstance Blueprint in the Hololight Stream VR Template keeps track of the connection state between Levels. This GameInstance type lets Blueprints get the current connection state. Select Project Settings > Maps & Modes > Game Instance Class to set this custom game instance.

Input mapping

The Unreal Engine Template has input mappings for the default devices it supports. These mappings are modified to leave out unsupported devices and add devices supported by Hololight Stream.

VRPawn

Unreal Engine's VRPawn, which is included in their default VR Template, handles most of the required functionality in Hololight Stream. This includes the main input actions, controller tracking, and more, but that isn't everything. In addition to this functionality, the Hololight Stream VR Template adds controller and hand visualization to the VRPawn and implements Hololight Stream's Connection State Handler and Controller State Handler interfaces to add logic for client and controller connection and disconnection. There are also examples of Hololight Stream types and functions inside the Blueprint. You can find more information by looking into the comments in the Blueprint itself.

Hololight Stream VT Template VRPawn Blueprint functions

Blueprint	Description
`void RegisterConnectionStateHandler(TScriptInterface<IStreamConnectionStateHandler> ConnectionStateHandler)`	Registers the object or actor which implements the IStreamConnectionStateHandler interface to the XR System's OnConnectionStateChanged event. This event returns the new connection state, which can be "Disconnected", "Connecting", "Connected", "Closing", and "Failed".
`void UnregisterControllerStateHandler (TScriptInterface<IStreamControllerStateHandler> ControllerStateHandler)`	Unregisters the object or actor which implements the IStreamControllerStateHandler interface from the XR System. This way the object will stop receiving the events.
`void RegisterControllerStateHandler (TScriptInterface<IStreamControllerStateHandler> ControllerStateHandler)`	Registers the object or actor which implements the IStreamControllerStateHandler interface to the XR Input System's OnControllerStateChanged event. This event returns the new state, which consists of the controller name, tracking status, controller type, and handedness.
`void UnregisterControllerStateHandler (TScriptInterface<IStreamControllerStateHandler> ControllerStateHandler)`	Unregisters the object or actor which implements the IStreamControllerStateHandler interface from the XR System. This way the object will stop receiving the events.

Widget menu

The Hololight Stream VR Template adds additional buttons to the Widget Menu in the VR Template. These buttons are for toggling passthrough, audio, microphone, and disconnection. These buttons demonstrate Hololight Stream features through Blueprint functions.

2.3.2. Additional setup

Maps & Modes

Select **Project Settings > Maps & Modes**. There you will find a Hololight Stream example game mode, Hololight Stream-ready maps for Unreal Editor, as well as Hololight Stream Game and Game Instance settings to be use in the project.

Rendering

Select Project Settings > Rendering. The VR Template comes with settings already set up for VR applications, but the template comes with higher-end settings in place. You can adjust these settings according to your hardware and application needs.
Use performance-demanding effects with care.

Instanced Stereo

Instanced Stereo sets Unreal to render stereo frames on the same pass. This helps with the performance of XR applications.[1] However, in some cases it can also cause issues with materials.

[1] https://dev.epicgames.com/documentation/en-us/unreal-engine/xr-performance-features-in-unreal-engine

Windows platform

Select Project Settings > Platforms > Windows to change settings specific to Windows. One important setting here is Default RHI (Render Hardware Interface). Hololight Stream currently supports DirectX 11 and DirectX 12, but there might be a difference in performance or render quality between the two.

For more information, check out Unreal's documentation about the **Render Hardware Interface (RHI)**.

Plugin settings

Select **Project Settings > Plugins > Hololight Stream Settings**, to see and change settings related specifically to Hololight Stream. Hololight Stream saves the settings in a custom configuration file. With this file, you can still modify port settings after packaging your application. If the configuration file is corrupted and the application can't parse it anymore, Hololight Stream Client will not be able to properly connect.

The configuration file is located at "HololightStream/Resources/remoting-config.cfg" in Unreal Editor or at "Config/remoting-config.cfg".

3. Use Hololight Stream

NOTE	Currently, Hololight Stream for Unreal Engine does not support depth transmission. Disabe this setting on the client.

3.1. Hololight Stream settings

To change Hololight Stream settings, select **Edit > Project Settings > Plugins > Hololight Stream Settings**.

Hololight Stream setting defaults

Setting	Default value
Signaling port	9999
Encoder Bandwidth (Kbps)	-1
Statistics Logging Enabled (Experimental)	Unchecked
Minimum Port	50100
Maximum Port	50100

3.2. Components

3.2.1. Connection

Disconnection

The EnableHMD function has an input boolean to use for disabling the HMD. The HMD is enabled automatically in the VR Preview start or when the Start in VR option is selected in a game build. Disabling the HMD via this function will close the connection and disconnect the client.

After that, it can be called again with true to re-enable the HMD and the connection will be reopened for a new client connection.

For more information on EnableHMD in Unreal, check out **EnableHMD**.

Hololight Stream connection Blueprint functions

Blueprint	Description
`bool GetConnectionInfo(FStreamConnectionInfo& ConnectionInfo)`	Returns true if there is a connection and false if not. Also returns connection information, which consists of remote (client) name, remote (client) version, rendering config, remote (client) device type and the codec in use.
`void RegisterConnectionStateHandler(TScriptInterface<IStreamConnectionStateHandler> ConnectionStateHandler)`	Registers the object or actor which implements the IStreamConnectionStateHandler interface to the XR System's OnConnectionStateChanged event. This event returns the new connection state, which can be "Disconnected", "Connecting", "Connected", "Closing", and "Failed".
`void UnregisterControllerStateHandler (TScriptInterface<IStreamControllerStateHandler> ControllerStateHandler)`	Unregisters the object or actor which implements the IStreamControllerStateHandler interface from the XR System. This way the object will stop receiving the events.

3.2.2. Input

The Motion Controller Component tracks controllers or hands in the level. Get Hand Tracking State Blueprint function can be used to get hand tracking data (including bone data). Get Motion Controller State Blueprint function can be used to get controller tracking data. For Unreal versions 5.4 and older, there is only Get Motion Controller Data function that combines these two functions.

Tracking

Hololight Stream provides tracked controller and hand data to the Unreal Engine's XR System. You receive the tracked poses of controllers or hands using Motion Controller Components or Blueprint functions Get Motion Controller State and Get Hand Tracking State. As motion sources, Hololight Stream supports Left, Right, LeftPalm, RightPalm, LeftAim and RightAim.

Controller State

By implementing and registering the Hololight Stream controller interfaces, users can get the controller state whenever there is a new state for any controller.

Hololight Stream controller Blueprint functions

Blueprint	Description
`void RegisterControllerStateHandler (TScriptInterface<IStreamControllerStateHandler> ControllerStateHandler)`	Registers the object or actor which implements the IStreamControllerStateHandler interface to the XR Input System's OnControllerStateChanged event. This event returns the new state, which consists of the controller name, tracking status, controller type, and handedness.
`void UnregisterControllerStateHandler (TScriptInterface<IStreamControllerStateHandler> ControllerStateHandler)`	Unregisters the object or actor which implements the IStreamControllerStateHandler interface from the XR System. This way the object will stop receiving the events.

3.2.3. Input mapping contexts

Hololight Stream Plugin adds the keys for supported device controllers and hand interaction keys to the Engine. You can map these keys to input actions in the standard input system or in Enhanced Input. For Enhanced Input, go to **Project Settings > Engine > Enhanced Input > Default Mapping Contexts** and add the Input Mapping Contexts you created. You can also add Input Mapping Contexts during runtime using the **Add Mapping Context** Blueprint function on runtime. You can use these Input Actions in the C++ code of Blueprints to assign logic to them.

3.2.4. Audio

To stream server audio to client device, you can use the provided Audio Blueprint functions:

Hololight Stream audio Blueprint functions

Blueprint	Description
`bool IsAudioEnabled()`	Returns true if the audio is streamed through Hololight Stream.
`bool ToggleAudio(bool bEnable)`	Enables or disables the audio streaming according to the input "bEnable". Returns true if the operation is successful.

3.2.5. Microphone

To capture microphone input from the client device and play it on the Unreal Engine Stream server, use Unreal Engine's Audio Capture Component the same way as you would for local applications.
To use the correct audio capture stream, modify the engine configuration. You can find this under Config/DefaultEngine.ini. Add the following lines to the configuration file:

```
[Audio]
AudioCaptureModuleName=StreamMicrophoneCapture
```

Then restart Unreal Engine.

The capture can be enabled before the connection and will start when the connection is established. Additionally, the Stream Audio Capture Extension Component needs to be used together with the Audio Capture Component for Hololight Stream to handle microphone stream (see below).

NOTE	Currently, Hololight Stream microphone capture only works with one Audio Capture Component in the level. Having multiple Audio Capture Components can lead to unexpected behavior.

Hololight Stream Audio Capture Extension Component

To securely handle audio streaming through Hololight Stream, you need the Hololight Stream Audio Capture Extension Component. You should create, start, stop, and destroy this component together with the Audio Capture Component. You can also bundle the two components together.

3.2.6. Passthrough

Hololight Stream supports passthrough mode for devices like Meta Quest 2, Meta Quest Pro, and Meta Quest 3 that support it.

Enable passthrough

To enable passthrough in Unreal Engine:

1. Select **Edit > Project Settings**,
2. Set **Alpha Output** to true.

To use passthrough on the Level in the VR Template, remove or hide the Sky Sphere.

NOTE	In Unreal Engine 5.4 and earlier Alpha Output is **Enable alpha channel support in post processing**. Set this to **Allow through Tonemapper**.

Use passthrough

In order to use passthrough during runtime, use the Blueprint functions to get and set the passthrough state. Additionally, for passthrough to work properly, the level's background should be be black or transparent. This means not using a Sky Sphere.

Example

To use passthrough the server-side, refer to the VR Template's WidgetMenu Blueprint. This sample demonstrates the two Hololight Stream functions related to passthrough:

Hololight Stream passthrough Blueprint functions

Blueprint	Description
`bool GetPassthrough()`	Returns true if the passthrough is enabled and false if not.
`bool TrySetPassthrough(bool bEnable)`	Enables or disables the passthrough according to the input "bEnable". Returns true if the operation is successful.

Additionally in the **VRPawn**, there is an example of **Get Passthrough** for getting the client setting on connection to setup the level.

Limitations

- Intermediate opacity values are only supported for black.
- Skyboxes with a solid color other than black may cause color bleeding on the borders of objects.

4. Package Project to build application

4.1. Disable Fab

Fab is the Unreal Engine asset store. By default, Unreal Engine has Fab enabled. If both Hololight Stream and Fab are enabled, the application won't work when you package it. To take care of this:

1. Select **Edit > Plugins**.
2. Select **ALL PLUGINS**.
3. Disable **Fab**.

4.2. Enable Start in VR

For Hololight Stream, we recommend starting VR rendering automatically. You should do this even if you plan to transition out of VR rendering and back into doing it at a later point in the application. You may have issues, but you also may not.

To do this, select **Project Settings > Description > Settings**. Enable **Start in VR**.

NOTE	To enable VR rendering manually, use the EnableHMD function. This function enables stereo rendering and Hololight Stream. This means a Hololight Stream connection can only be established when stereo rendering is enabled. You can use this function in both Blueprints and C++ code. For more information on EnableHMD, see **Disconnection.**

4.3. Game Default Map

Before building the application, you need to set the Level you want to use as the Game Default Map so

To package the project, select **Platforms > Windows > Package Project**.

NOTE	If you need to make changes to your packaging configuration, go to **Platforms > Windows.**