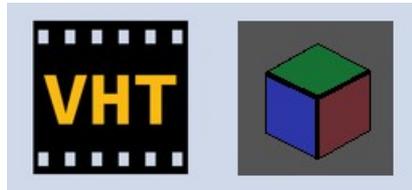


VHT tools for DAZ

User Guide

v1.0



virtualhometheater.com

Table of Contents

Introduction.....	3
Concept Preview.....	4
Use Case Examples.....	5
Software Requirements.....	5
Hardware Requirements.....	6
Required Work Style.....	6
Highlighted Features.....	6
Installation.....	7
Prerequisite Settings.....	8
Daz Studio Preparation.....	9
Quick First Use.....	12
VHT Tools Interface Description.....	13
Cameras.....	13
Add_camera_bundle.duf.....	15
Add_camera_EQD180.duf.....	15
Add_camera_EQDpivot.duf.....	15
Add_camera_REL9mm.duf.....	15
Add_camera_RELphoto.duf.....	15
Add_camera_VR180.duf.....	16
Add_camera_VR360.duf.....	16
Default Start Presets.....	16
default_preset_#0.dse.....	16
default_preset_#1.dse.....	16
default_preset_#2.dse.....	16
Layout_SbS.dse.....	16
Camera Director.....	17
Use_the_selected_stereo_camera_for_viewport.dse.....	17
Current_viewport_camera_favorite_pose_get.dse.....	17
Current_viewport_camera_favorite_pose_set.dse.....	17
Current_viewport_camera_lock_pose_off.dse.....	17
Current_viewport_camera_lock_pose_on.dse.....	17
Repose_VR_viewport_camera_as_selected_camera.dse.....	18
Move_selected_camera_to_selected_node.dse.....	18
Set_camera_name.dse.....	18
Set_rendering_pixel_size_for_selected_camera.dse.....	18
Drawing Settings.....	19
DrawStyle_FILAMENT.dse.....	19
DrawStyle_NVIDIA_Iray.dse.....	19
DrawStyle_LINE.....	19
DrawStyle_TextureShaded.dse.....	19
DrawStyle_WireBoundingBox.dse.....	19
QuickDrawMode_box.dse.....	19
QuickDrawMode_hiddenlines.dse.....	20
Floor_off.dse.....	20
Floor_on.dse.....	20
Render Presets.....	20
RS_preview_high.duf.....	20
RS_preview_low.duf.....	20

RS_rendering.duf.....	20
Daz Studio to VHT messaging bridge.....	21
Launch_VHT_app.dse.....	21
Rendering.....	22
Render_selected_cameras.dse.....	22
VHT Control.....	26
VHT_desktops-VR_View.dse.....	26
VHT_gui_repose.dse.....	26
VHT_gui_switch.dse.....	26
VHT_hmd_repose.dse.....	26
VHT_level_update (tilted view).dse.....	26
VHT_level_zero (front view).dse.....	26
Miscellaneous.....	27
SAVE_incr.dse.....	27
Audio.....	27
Add_object_audio.dse.....	27
The workflow process for VR art creation with VHT tools.....	28
How to view the final rendered images or videos in VR.....	30
Working with audio objects.....	31
Visual Quality and Performance Evaluation.....	33
1 st Art preview in VR.....	33
2 nd Rendering the final VR art.....	34
3 rd Consuming the VR art.....	34
Demo samples.....	34
Acronyms.....	35
Links.....	36
Revision History.....	36

Introduction

“VHT tools for DAZ” is a set of scripts, presets and cameras for Daz Studio 3D modeling software specially designed as a concept (lets call it “Spatial Modeling”) to help CG artists to create 3D stereoscopic 180 or 360 degree panoramic computer generated art images or videos for VR users.

These tools will use the main viewport to draw a stereoscopic Sbs 180 or 360 degree projection for live scene view in VR and will help artist to render stereoscopic VR images or videos files.

To view in VR the live stereoscopic viewport, will require to use the Virtual Home Theater (VHT) VR application from Steam store and to have two displays: one for Daz Studio GUI and the second for stereoscopic viewport!

The render tool can be used also without the VHT VR application from Steam store or dual displays!

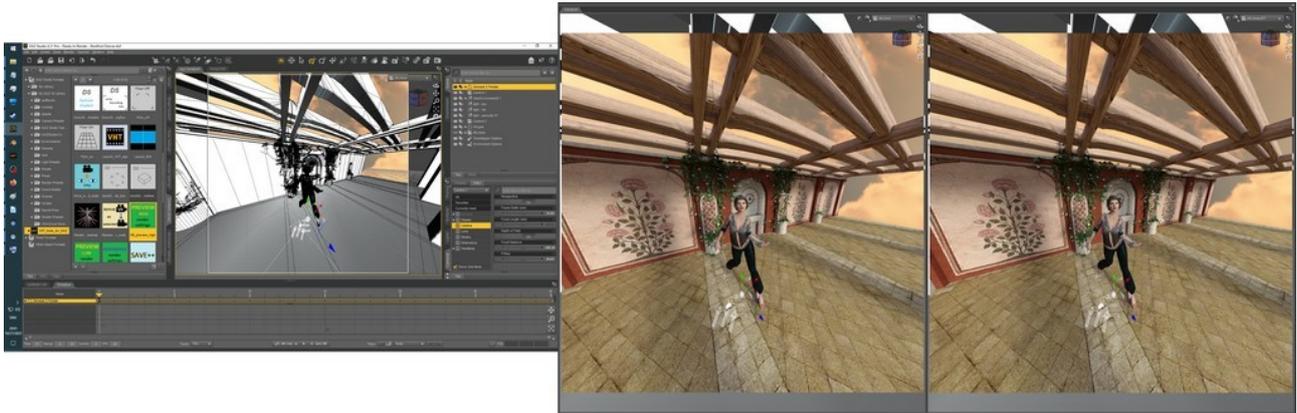
When using these tools, the artist will work as usual in front of his computer display and will use the VR headset and VHT VR application when a VR preview is needed, in a sited position using his mouse, keyboard and a SpaceMouse.

In VR the artist will see the main desktop screen where will be his Daz Studio GUI main window and surrounded by his ‘true to scale’ 3D scene, rendered live as stereoscopic 3D 180 or 360 degree panorama using any render engine available from Daz Studio: NVIDIA Iray (real-time ray-traced), Filament (PBR), etc.

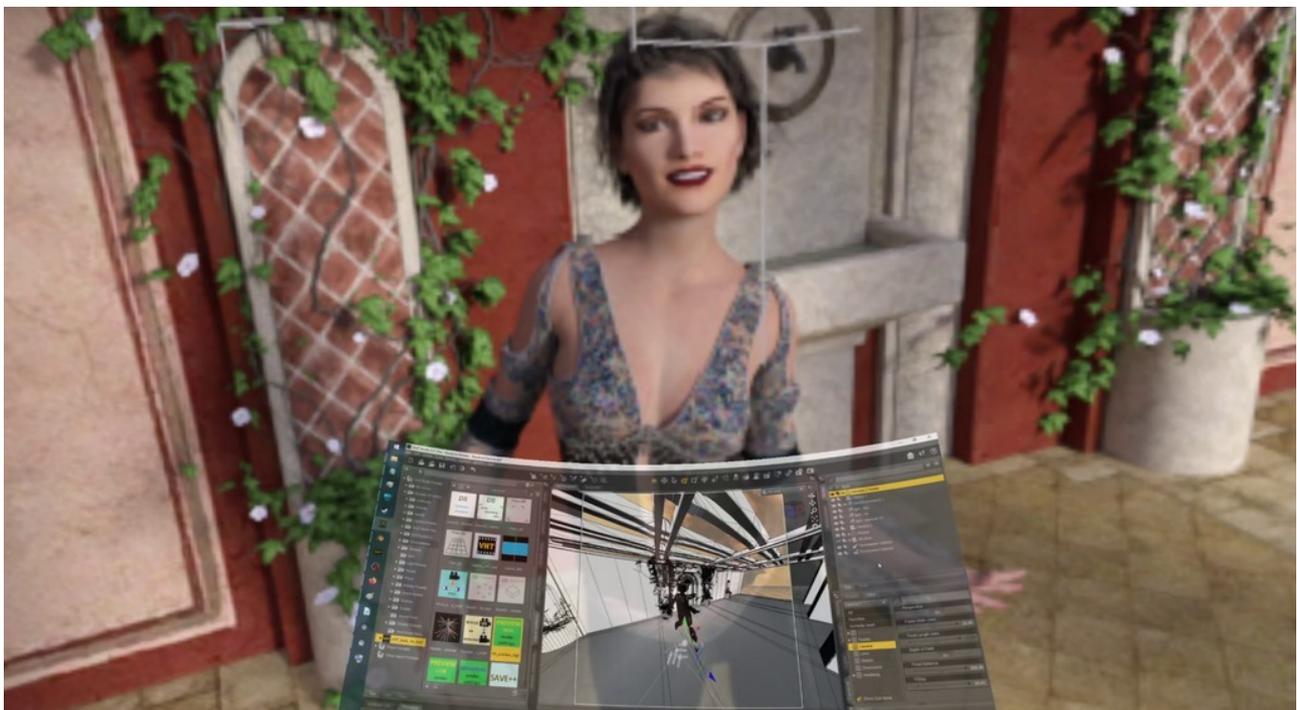
VHT tools can be used by a VR CG artist:

- for live stereoscopic scene preview in VR before final rendering
- to render final stereoscopic images or videos for publishing
- to check objects geometry for proportions and dimensions
- to check for 3D disparity and select the best IPD for stereoscopic cameras
- for 3D modeling, morphing and posing
- to check scene composition and animations
- for lighting setup, colors, materials, texture, environment and tone mapping tweaking
- to find the best scene view for VR (camera angle)
- to select the best lens projection/mapping
- to add audio objects

Concept Preview



This image is a desktop snapshot demo sample of what a user will see on his dual displays.



This image is a VR snapshot demo sample of what a user will see through the HMD lens.

Scene source:
DAZ Studio Tutorials/Ready to Render - Barefoot Dancer

The same concept product is also available from VHT for Blender 3D modeling software.

Use Case Examples

An artist can use these tools to create images or videos in stereoscopic 3D format for VR that are more immersive and spectacular than a classic presentation on a 2D flat surface (paper or screen) for:

- Architectural interior/exterior design presentation
- Vehicle (car, airplane, yacht etc.) interior/exterior design presentation
- Cloth design on a character for fashion presentation
- Furniture design presentation
- Engineering design evaluation
- Epic VR dioramas of historic events
- Visual novels for VR users
- re-rendering of old 3D scenes for VR users
- VR movies creation
- a game or movie character showcase

Software Requirements

- Windows 10
- **Daz Studio** 4.21 (for 3D modeling and rendering)
- **VHT tools for DAZ**
- **VHT** 1.18.0 (only for live stereoscopic scene preview in VR from Daz Studio and to view the final rendered images or videos)
- **FFmpeg** (used by VHT render tool script to create videos from the rendered animated image series)

Go to:

daz3d.com

virtualhometheater.com

ffmpeg.org

Hardware Requirements

The following hardware is required only when using VHT VR app for VR preview!

- **a 2nd display** (real, dummy or virtual; **2K** or **4K** native/custom) for the VR stereoscopic Viewport window
- **VR HMD** (high PPD PC models like **HP G2** or better for clarity)
- **NVIDIA RTX GPU** (the bigger VRAM the better, **4090** for performance and fast renders)
- **SpaceMouse** for 3D scene navigation and for high Viewport update rate

Remarks:

- **If you have only one display you will not be able to start VHT in Spatial Modeling mode.**
- The VR controllers are not required and should not be used.
- The SpaceMouse is required for ease of navigation through the scene and can be ignored if you use other preferred input methods.
- **If you don't have a SpaceMouse and you use the mouse to rotate and move then the frame update rate for Viewport will be low (<https://youtu.be/FYvjggqFwvs>).** To have a smooth high frame update rate for Viewport you should use a SpaceMouse. This is because of how Daz Studio update the Viewport and has nothing to do with the GPU performance or VHT.
- The 2nd display resolution will define the image clarity (visual PPD) for the live stereoscopic scene preview in VR.

Optional: **headphones** for binaural music listening from VHT when creating art.

Required Work Style

You will need to use the Aux Viewport as a main Viewport because the real main Viewport will be used for live 3D stereoscopic scene preview in VR.

Highlighted Features

- 8 stereoscopic camera types in 7 groups
- 3 lens projection/mapping models
- a render manager to render final VR images or videos for multiple camera selection
- control from DS the VHT virtual displays
- start VHT app and create an IPC between VHT app and 'VHT tools for DAZ' script
- add Audio Objects
- 24 miscellaneous tools

Installation

Unzip the “VHT tools for DAZ.zip” file.

Method 1:

Add a Base Directory... then chose the unzipped “VHT tools for DAZ” folder.



Method 2:

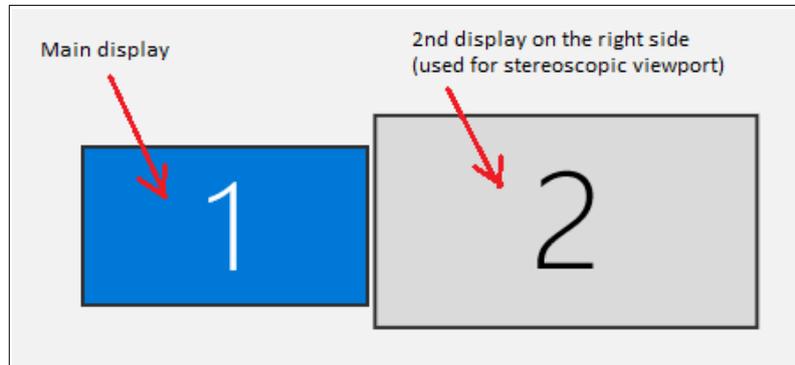
Copy “VHT tools for DAZ” folder and “VHT tools for DAZ.png” to “Documents\DAZ 3D\Studio\My Library”



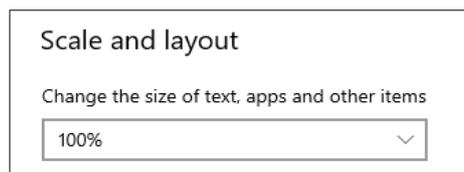
Prerequisite Settings

Set the display with the smallest resolution as the main display (recommended).

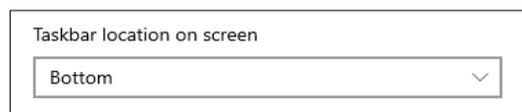
Rearrange your displays like shown below:



Set the scale for the 2nd display to **100%** (very important requirement)



For the 2nd display set the taskbar location on screen to **Bottom** (very important requirement)



If the second display has a lower native resolution than 2K (e.g. 1920x1080) you can try to change resolution by using the [“NVIDIA Control Panel > Display > Change resolution > Customize.. >Create Custom Resolution...”](#) type 2560x1440 or 3840x2160, then chose the new resolution from Display Settings.

Daz Studio Preparation

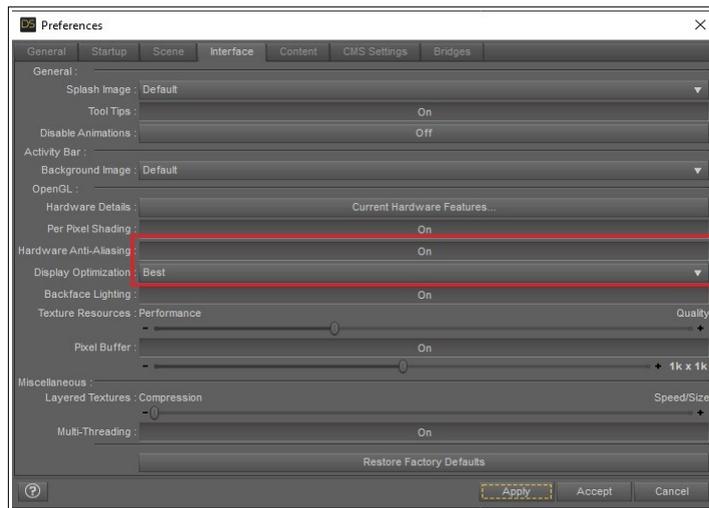
The main window from Daz Studio app must be placed on the main display.

Set the Layout to “City Limits Lite” and Style to “Darkside”.

Set the Preference/Interface settings as follow:

Hardware Anti-Aliasing = **On**

Display Optimization = **Best** (very important)



All the next steps are necessary because we can't create another Viewport and because on the Aux Viewport you can't change the layout.

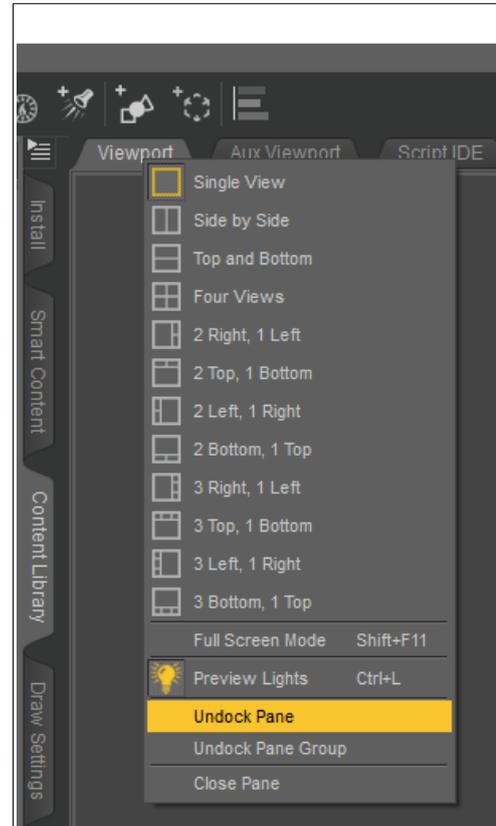
In order to create a stereoscopic Side-by-Side Viewport that will be used by VHT app to draw the scene in VR it is necessary to **undock and refit the Viewport on the 2nd display** (<https://youtu.be/CTGZT8KwNMI>).

You will use the “**Aux Viewport**” as a main Viewport on Daz Studio window.

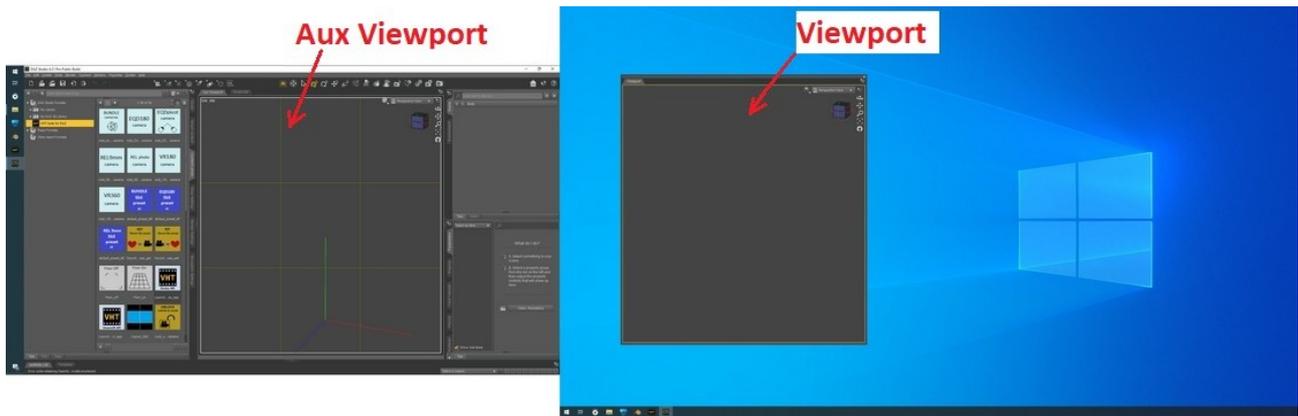
Do not use Shift+F11 for full screen or try to fit the window by dragging the Viewport's edges with mouse.

The next steps (done only once) are necessary because there is no maximize button available on this window frame.

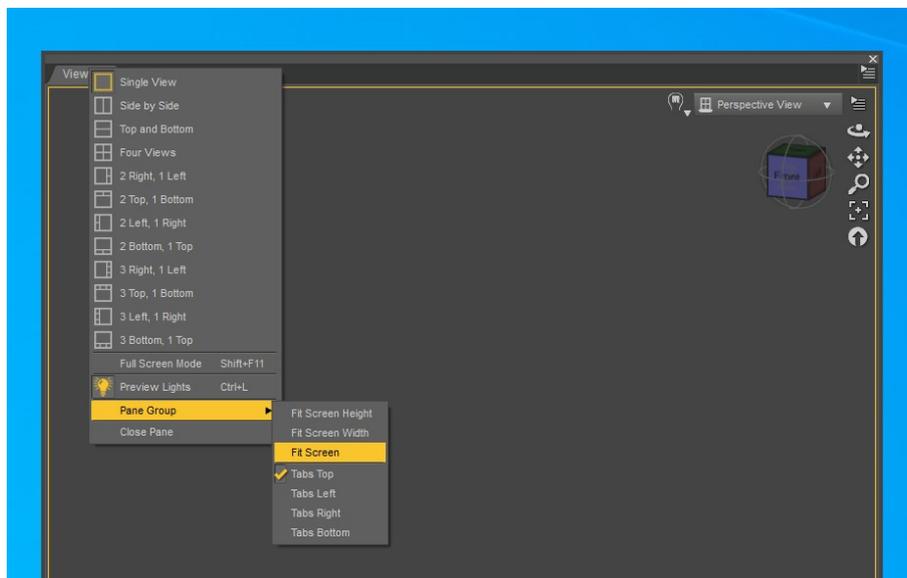
Right click on the Viewport and then “**Undock Pane**”



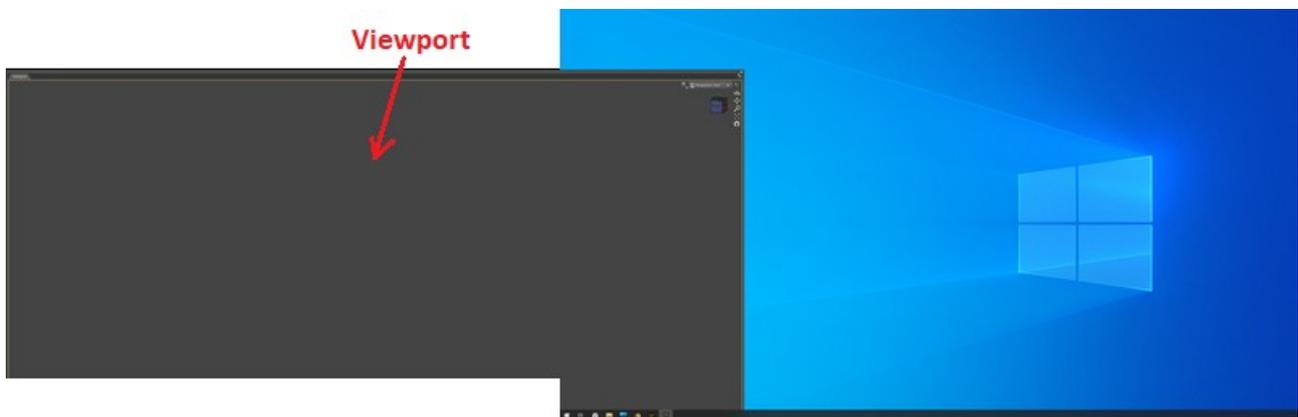
Move the undocked Viewport from the 1st display to the 2nd display then **move** the Aux Viewport from the right pane group to the center pane group.



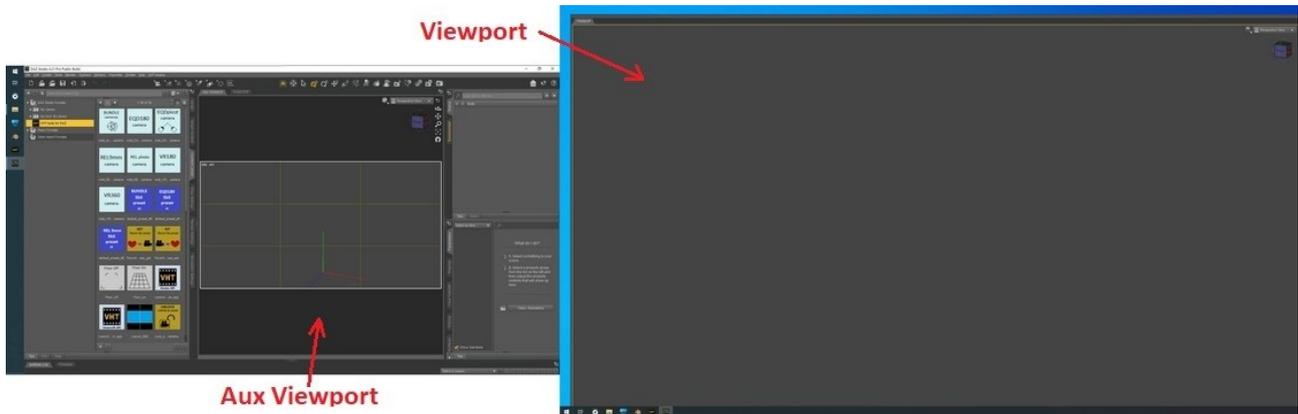
Make sure that for Viewport (not Aux Viewport) **“Tabs Top”** is selected then click on **“Fit Screen”**:



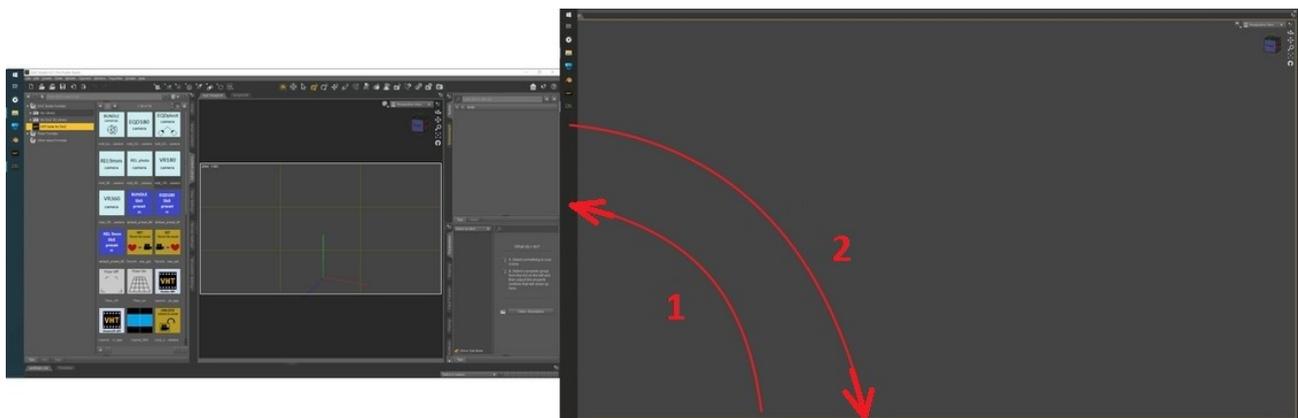
Then, you should see something like this:



Move again the undocked Viewport from 1st display to the 2nd display like shown below near to the upper left corner of the 2nd display:

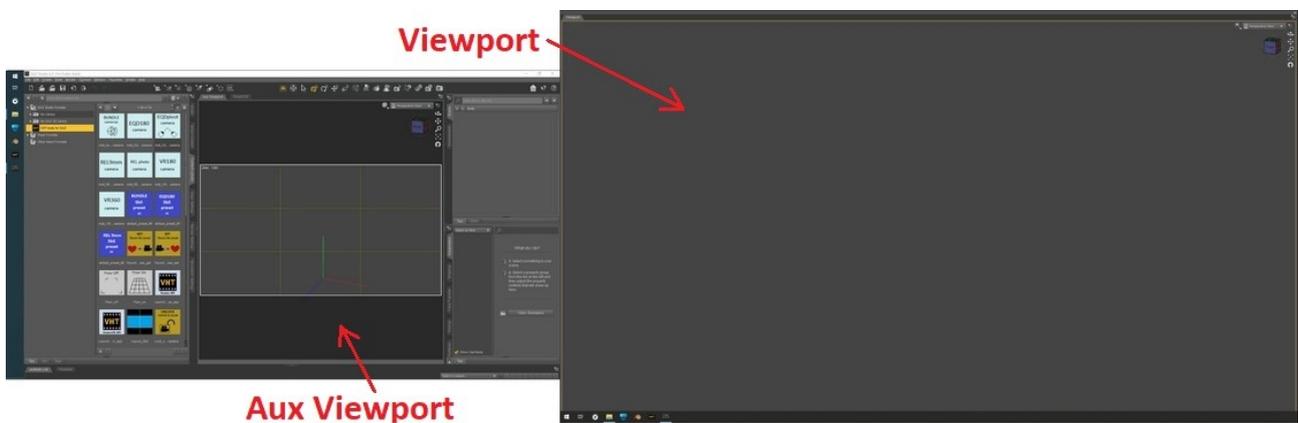


Drag the Taskbar from the 2nd display to the Left side then drag-it again to Bottom like shown below:



By doing this the Viewport will self resize and fit exactly to the 2nd display.

The final layout result should look like this:



Done.

Now you are ready for Spatial Modeling.

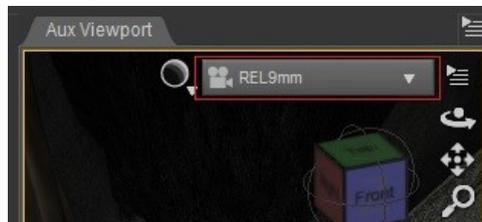
Quick First Use

Assuming you have done the prerequisite preparations as described above, do the next steps to start a VHT-Daz Studio VR work session:

1. Log-in to your Steam account.
2. Turn On your VR HMD.
3. In Daz Studio load the scene you want to see/work in VR.
4. Double-click on:



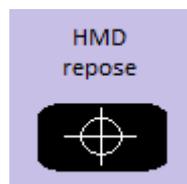
5. Select the REL9mm stereoscopic VR camera for Aux Viewport:



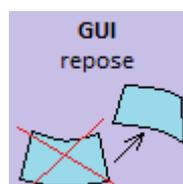
6. Start VHT from Daz Studio (not from Steam library):



7. Put your VR HMD on your head.
8. If necessary use your mouse to bring Daz Studio window to front. Use the mouse and keyboard to interact with Daz Studio as before but in VR.
9. If necessary use



and/or



to recenter HMD and GUI in VR.

10. Use your **SpaceMouse** to navigate through scene. If you don't have a SpaceMouse read the remarks from Required Hardware chapter above.

11. You can change the render engine for VR to



VHT Tools Interface Description



Do not rename the scripts, cameras and render presets or move them to a subfolder.

Cameras

All VR cameras (except VR360_MONO) are stereoscopic with an IPD of 65 mm as default except for VR180, VR360, EQD180_IPD10 and EQD180_IPD40.

The user can partially rename a camera using the VHT tool *Set_camera_name* script.

The user can change the IPD of each camera using the VHT tool *Set_camera_IPD* scrip.

A match between the human eye IPD and the camera IPD is not necessary (use what is best for your scene context):

- Use a lower IPD than 65 mm for close objects or 360 panoramic views.
- Use very large IPD (meters) if you want to have a toy/miniature-like view renders.
- VR180 and VR360 cameras have low IPD for a comfortable viewing experience.

DS use two render engines models to display 3D objects:

- OpenGL (rasterized rendering) only for rectilinear projection/mapping
- NVIDIA Iray (ray-traced rendering) for all projection/mapping types: rectilinear, equidistant, spherical, etc.

When scene is changing, DS will use the quick draw mode that use the OpenGL render engine even if the camera is set to NVIDIA Iray render engine.

Equidistant and Spherical Equirectangular Panorama projection/mapping will be used only when using NVIDIA Iray render engine and the view is not changing, otherwise the camera is set to render using OpenGL rectilinear 9mm focal lens.

To avoid projection/mapping mismatch between VHT and DS, the *Launch_VHT_app* script will update VHT VR view projection/mapping to be the same as to what DS use at any moment in real time without the user intervention. This is done by using the messaging bridge (IPC) created by *Launch_VHT_app* script.

Each camera group has a SpotLight turned On as default to make sure user will see something if the scene has no light, so you need to make sure the SpotLight is Off when you do final render if you don't needed.

You can filter the stereo camera lights by using "SCL_" keyword.

Do not use the DS camera's Headlamp for stereoscopic rendering, the render will have 3D stereoscopic disparity shadows. If you need a headlamp in the final render use the SCL_SpotLight from the camera group. There will be no 3D disparity shadows when using SCL_Lights.

If the scene is dark when using the NVIDIA Iray render engine increase the Film ISO from Render Settings/Tone Mapping.

Each camera group has a root camera and children cameras.

The children cameras are fixed to the root camera and can't be used to rotate and translate the view.

For view rotation and translation use the root camera by selecting it from the predefined view cameras list from the Aux Viewport.

Rotation and translation of a camera view should be done from the Aux Viewport.



If the Viewport has a root camera then you can use the left frame to do camera view rotation and translation.

On Viewport you can use any root or children type stereoscopic camera by using the VHT tool *Use_the_selected_VR_camera_for_viewport* script to make a selected camera active for VR view.

If NVIDIA Iray is not used on Viewport (by DS or user) then cameras like EQD180, VR180 or VR360 are set to render the scene using the rectilinear 9 mm focal length OpenGL raster projection/mapping (127 degree FoV).

Add_camera_bundle.duf



Add a group of multiple cameras to your scene: REL9mm, EQD180, EQD180_IPD10, EQD180_IPD40, VR180, VR360 and VR360_MONO.

This group is designed to be used when working on your scene.

Because the group cameras have the same view pose vector you can switch between different type of cameras to evaluate the camera type you need for a scene view.

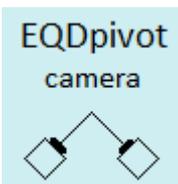
Add_camera_EQD180.duf



Add one stereoscopic camera with an Equidistant projection/mapping lens.

IPD = 65 mm, FoV = 180 degree, valid only for NVIDIA Iray

Add_camera_EQDpivot.duf



Add a stereoscopic camera with an Equidistant projection/mapping pivot lens (as the human vision but not necessary good for VR use).

IPD = 65 mm, FoV = 180 degree, valid only for NVIDIA Iray

Add_camera_REL9mm.duf



Add one stereoscopic camera with a Rectilinear projection/mapping lens.

IPD = 65 mm, focal length = 9 mm, FoV = 127 degree, valid for any render engine.

Add_camera_RELphoto.duf



Add one stereoscopic standard rectilinear camera.

IPD = 65 mm, focal length = user defined, valid for any render engine.

You can use this camera to render a stereoscopic photo or a poster for VHT environment.

Add_camera_VR180.duf

VR180
camera

Add one stereoscopic camera with a Spherical Equirectangular Panorama projection/mapping lens.

IPD = 6.5 mm, FoV = 180 degree (is a crop of a VR360 camera), valid only for NVIDIA Iray

Add_camera_VR360.duf

VR360
camera

Add one stereoscopic and one mono panoramic camera with a Spherical Equirectangular Panorama projection/mapping lens.

IPD = 6.5 mm, FoV = 360 degree, valid only for NVIDIA Iray

Default Start Presets

default_preset_#0.dse

BUNDLE
SbS
preset
#0

Create a SbS split Viewport layout, add the bundle cameras group, set the Texture Shaded draw style and use the render settings preview_low.

Use this when you create a new scene or load a scene without a VR camera.

default_preset_#1.dse

EQD180
SbS
preset
#1

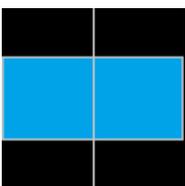
Create a SbS split Viewport layout, add the EQD180 camera, set the NVIDIA Iray draw style and use the render settings preview_low.

default_preset_#2.dse

REL 9mm
SbS
preset
#2

Create a SbS split Viewport layout, add the REL9mm camera, set the Filament (PBR) draw style and use the render settings preview_low.

Layout_SbS.dse



Create a SbS split Viewport layout.

Camera Director

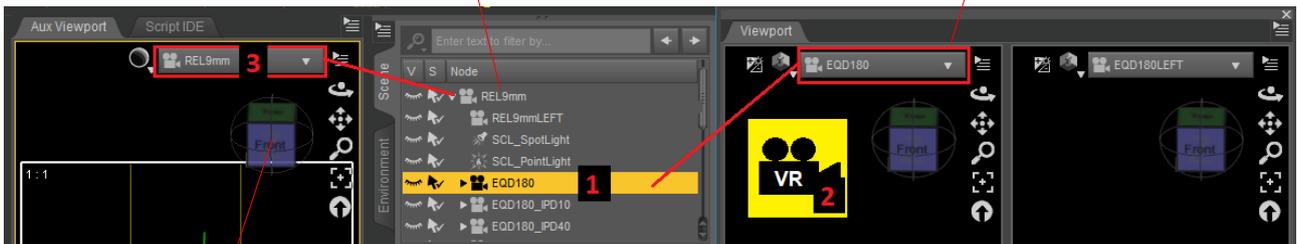
Use_the_selected_stereo_camera_for_viewport.dse



Use the selected stereo camera (root or children) for stereoscopic Viewport.

Use this to set the VR camera that will be used for live stereoscopic scene preview in VR (see step 1 and 2 from below image).

To be able to change the view pose of the **stereo camera used for Viewport** you need to manually select the **root camera** by selecting it from the predefined view cameras list from the Aux Viewport (step 3).



Then, by using this box or the SpaceMouse you can change the pose of the VR camera.

Current_viewport_camera_favorite_pose_get.dse



Recall the saved favorite pose for the current Viewport stereoscopic camera.

Current_viewport_camera_favorite_pose_set.dse



Save the current Viewport stereoscopic camera pose as favorite.

Current_viewport_camera_lock_pose_off.dse



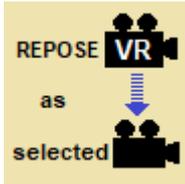
Unlock the current Viewport stereoscopic camera pose.

Current_viewport_camera_lock_pose_on.dse



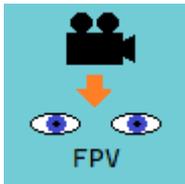
Lock the current Viewport stereoscopic camera pose.

Repose_VR_viewport_camera_as_selected_camera.dse



Repose the current Viewport stereoscopic camera as the selected camera or node.
Use this if you want a camera to have the same view as any DS camera from a scene.

Move_selected_camera_to_selected_node.dse

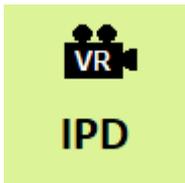


Move any selected camera to a selected node.
If the node name is “Head” then the stereoscopic camera will be placed as to have the same view as the character (FPV).

Use this if you want to render an image of a character point of view.

Because the moved camera will be the child of the head node it will follow the character movement and head orientation.

Set_camera_IPD.dse



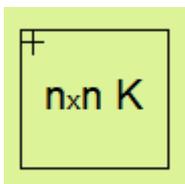
Change the current Viewport stereoscopic camera IPD.

Set_camera_name.dse



Rename the current Viewport stereoscopic camera.

Set_rendering_pixel_size_for_selected_camera.dse



Change the rendering (not the live preview) pixel size for the selected stereoscopic camera.

Default size = 4096x4096 for 180 degree and 8192x4096 for 360 degree cameras.

Drawing Settings

These scripts are changing the stereoscopic Viewport draw style that is used only for live stereoscopic scene preview in VR.

The Quick draw mode is active when you change the view pose (vantage point).

DrawStyle_FILAMENT.dse



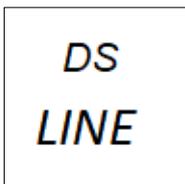
Set the VR draw style to Filament (PBR).

DrawStyle_NVIDIA_Iray.dse



Set the VR draw style to NVIDIA Iray.

DrawStyle_LINE



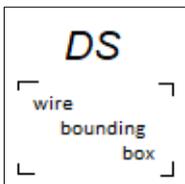
Set the VR draw style to Hidden Line.

DrawStyle_TextureShaded.dse



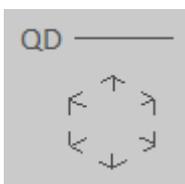
Set the VR draw style to Texture Shaded.

DrawStyle_WireBoundingBox.dse



Set the VR draw style to Wire Bounding Box

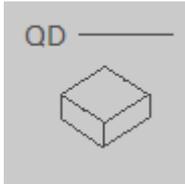
QuickDrawMode_box.dse



Set the VR quick draw mode to box.

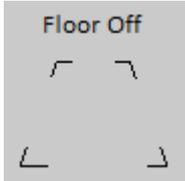
Use this if you have complex scene and a low performance GPU.

QuickDrawMode_hiddenlines.dse



Set the VR quick draw mode to hidden lines (DS default).

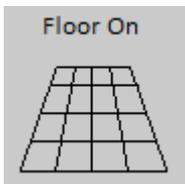
Floor_off.dse



Turn Off the floor in VR.

Not available when using NVIDIA Iray draw style.

Floor_on.dse



Turn On the floor in VR.

Not available when using NVIDIA Iray draw style.

Render Presets

RS_preview_high.duf



Set the render settings to:

MAX samples = 500, Denoise = On

Use this to preview your scene in VR or for final scene rendering.

RS_preview_low.duf



Set the render settings to:

MAX samples = 32, Denoise = On

Use this to preview your scene in VR (default).

RS_rendering.duf



Set the render settings to:

MAX samples = 15000, Denoise = Off, Rendered Converged Ratio = 99%,

Max Time = 0 sec. Use this for final scene rendering (long rendering time).

For more info about render settings read:

<https://blog.daz3d.com/best-iray-settings-for-daz-studio/>

Daz Studio to VHT messaging bridge

To view the stereoscopic SbS Viewport in VR it is necessary to start the VHT VR app in desktop mirror spatial modeling mode using the *Launch_VHT_app.dse* script tool to create a messaging bridge (IPC) between DS and VHT.

This bridge will make VHT know when drawing style and image projection/mapping are changing on DS Viewport to change the used VHT projection accordingly.

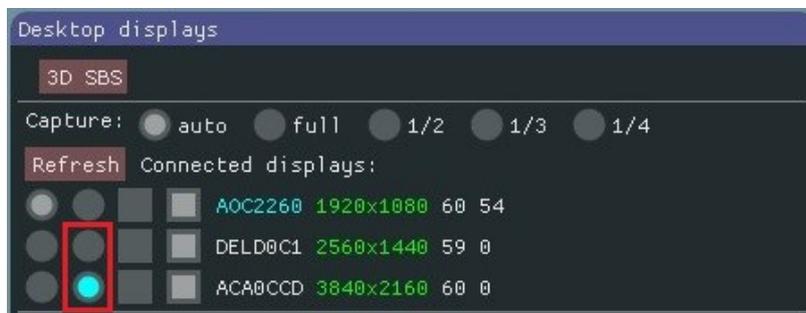
The messaging bridge will be used also to send user requested commands to VHT from DS using the following tool scripts:

- VHT_desktops-VR_View
- VHT_gui_repose
- VHT_gui_switch
- VHT_hmd_repose
- VHT_level_update (tilted view)
- VHT_level_zero (front view)

VHT app will be used by user to view in VR the stereoscopic SbS Viewport as a surround scene environment with an overlay main desktop display (DS GUI).

When VHT is started it will chose automatically the highest resolution display as the source for the VR stereoscopic Viewport image.

You can change the source of stereoscopic image to be another display (if you have more than two) by using the VHT UI:



Launch_VHT_app.dse



This will launch VHT app in desktop mirror spatial modeling mode and will create a messaging bridge (IPC) between DS and VHT.

Double-click again to end the VR session and close VHT app.

If the scene has no stereoscopic camera then this script will add one REL9mm camera.

Rendering

Before rendering it is recommended to end any VR application that is using GPU resources required for rendering.

Check list before final rendering:

- Make sure the VR cameras spotlights are turned Off (if you don't need them)
- Check the image format for rendered images: PNG or JPG
- If your render will take many hours move DS to another Windows Desktop using Win+Tab

These rendered images are using the cross-eye format so you can preview them with your own eyes without any VR device using the cross-eye method.

Rendering can be canceled anytime by pressing the CANCEL button on progress window.

Render_selected_cameras.dse



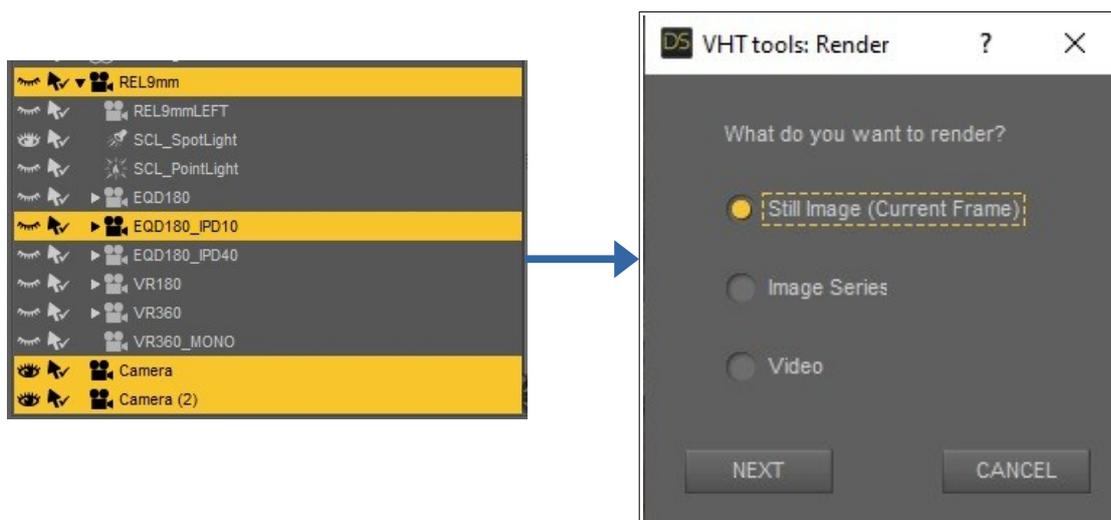
Launch a script that will manage the rendering of multiple selected cameras as still image (current frame), image series or video.

The render engine used will be NVIDIA Iray (MDL).

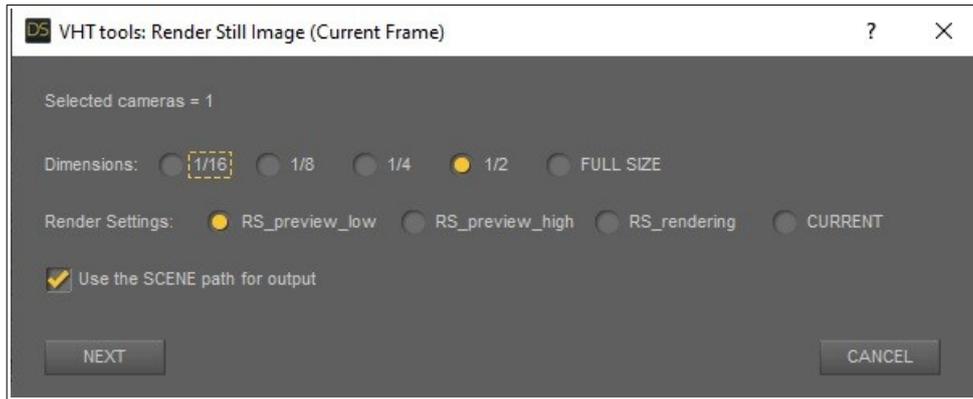
Any camera can be selected: normal DS cameras or stereoscopic cameras.

Select the VR360_MONO camera when you want to render a non-stereoscopic 360 panoramic image.

The render manager will display a dialog windows to chose the render type you want to do.



If you click NEXT the script will try to close the running VR applications to free GPU resources.



FULL SIZE dimension is the camera pixel size (default is 4096x4096) and should be used for final renders.

For quick preview renders select a lower fraction (e.g. 1/2 is half the camera pixel size = 2048x2048 resulting in a 4x faster renders)

RS_preview_low will use: MAX samples = 32, Denoise = On

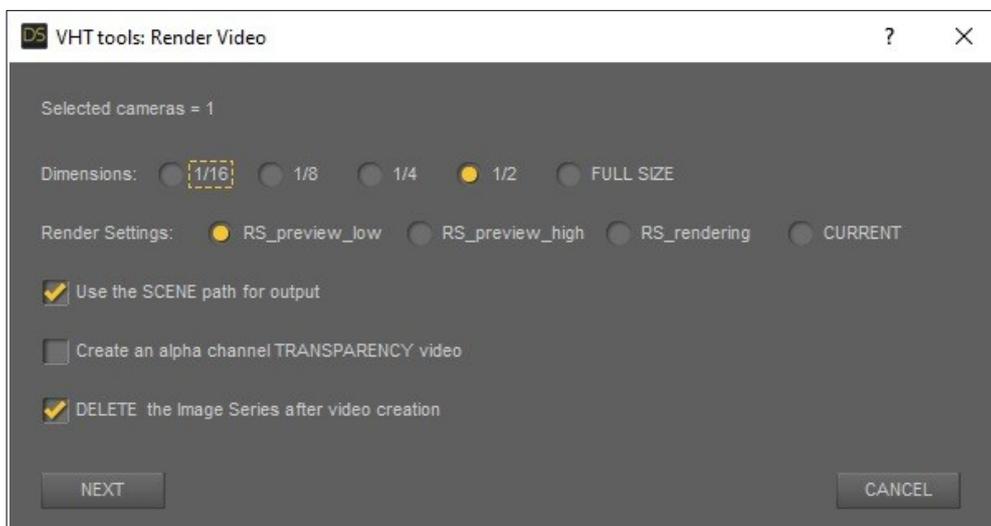
RS_preview_high will use: MAX samples = 500, Denoise = On (for quick final renders)

RS_rendering (recommended for professional final renders) will use: MAX samples = 15000, Denoise = Off, Rendered Converged Ratio = 99%, Rendering Quality = 1, Max Time = 0 sec

CURRENT will use the current user defined render settings.

Use the SCENE path for output

will save the render images or videos to the scene folder, otherwise the saving path will be the one defined in the render settings: Image Path for images and Series Path for image series and videos.



Create an alpha channel TRANSPARENCY video will encode the video using the VP9 codec.

DELETE the Image Series after video creation, if Off, no video frame image will be deleted.

This render manager script will use the Image Name or Series Base from DS/Render Settings/General for saving the rendered images or videos using the following format:

SceneName R_Date-Time_CameraName_Format_TiltAngle_.FileType

Example: “Ready to Render - Barefoot Dancer R_2023-10-26-12-54-7_EQD180_FF3D_SbS_crosseye_tilt-9_.png”

This render manager script will not use the built-in video render from DS but instead will execute an ffmpeg.exe command to create a video from rendered image series.

The FFmpeg input arguments used to encode a video are:

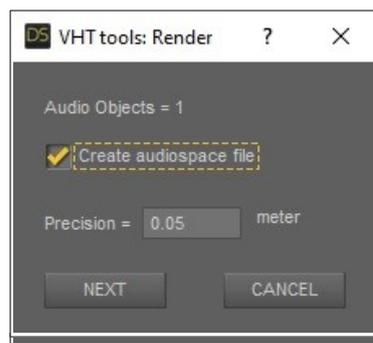
- when using alpha transparency: `-c:v vp9 -b:v 0 -crf 23 -pix_fmt yuva420p`
- for non-transparent video: `-c:v hevc_nvenc -b:v 0 -crf 23`

If you want to create a video using your own preferred input arguments or use another software to create a video from images then chose Image Series or Video but turn Off “DELETE the Image Series after video creation” to create the video for preview purpose.

For FFmpeg commands to work you need to add the FFmpeg folder path to “Windows_10 > System Properties > Advanced > Environment Variable > User variables for admin > Path > Edit > New”.

If you add an audio objects to your scene an audiospace companion file will be created for each rendered still image or image series / video.

Precision is how much an audio object must change its position reference to camera space to record the change on this file. For more info about audiospace visit VHT website to download the ‘VHT Audiospace - User Guide’.



```
Ready to Render - Barefoot Dancer R_2023-10-26-12-54-7_EQD180_FF3D_SbS_crosseye_tilt-9_audiospace - Notepad
File Edit Format View Help
VIRTUAL_HOME_THEATER_AUDIOSPACE_FORMAT_VERSION 1

This audiospace companion file is used by 'Virtual Home Theater' (VHT) VR multimedia player software to render
3D audio for CG VR videos or images, and was generated by 'VHT tools for DAZ - Render_selected_cameras' script
tool version 1.0.0

track=0
mode=STATIC_AUDIO_OBJECTS
name=bird

channel[0]*dB(0)->xyz(0.010,0.025,0.708):A0 bird
```

The render manager script will signal each rendering step with an acoustic signal (beep).

When rendering is done a rendering log file will be created as a companion file for each rendered still image or image series / video:

```
Ready to Render - Barefoot Dancer R_2023-10-26-12-54-7_EQD180_FF3D_SbS_crosseye_tilt-9_renderinglog - Notepad
File Edit Format View Help
RENDERING LOG

Generated by 'VHT tools for DAZ - Render_selected_cameras' script tool version 1.0.0

Image rendered with DAZ Studio version 4.21.0.5
Render engine: NVIDIA Iray
Hardware device(s):
#0 : NVIDIA GeForce RTX 4090
#1 :

Camera (projection) : EQD180
Format : _FF3D_SbS_crosseye_
Tilt = -9 degree

Image resolution = 8192x4096
Image FoV = 180 degree
Image PPD = 23
Image IPD = 65.0 mm
Left stereo offset = -65.0 mm
Right stereo offset = 0.0 mm

Max Samples = 15000
Max Time (secs) = 0
Rendering Quality Enable = true
Rendering Quality = 1
Rendering Converged Ratio = 0.990
Post Denoise = false

Total time (minutes) = 7.967

Dimensions = FULL SIZE
Render preset used: RS_rendering
```

The output files after a render job is finished:

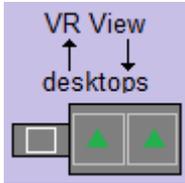
Name	Type	Size
Ready to Render - Barefoot Dancer R_2023-10-26-12-54-7_EQD180_FF3D_SbS_crosseye_tilt-9_audiospace	File	1 KB
Ready to Render - Barefoot Dancer R_2023-10-26-12-54-7_EQD180_FF3D_SbS_crosseye_tilt-9_.png	PNG File	49,162 KB
Ready to Render - Barefoot Dancer R_2023-10-26-12-54-7_EQD180_FF3D_SbS_crosseye_tilt-9_renderinglog	RENDERINGLOG F...	1 KB
Ready to Render - Barefoot Dancer.duf	DSON User File	135 KB
Ready to Render - Barefoot Dancer.png	PNG File	23 KB

Then by using the built-in image viewer from VHT VR app you can view the final rendered images or videos.

VHT Control

These control scripts will work only if DS to VHT messaging bridge is created by launching VHT VR app with Launch_VHT_app.dse script from DS.

VHT_desktops-VR_View.dse

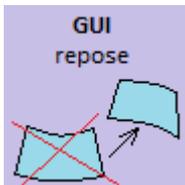


Switch between VR view and desktop view mode for screens.

VR view (default spatial modeling mode) will display the stereoscopic Viewport from DS as a 3D sphere projection/mapping.

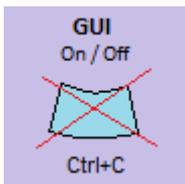
Desktop view will show all desktops as flat screens in front of the user.

VHT_gui_repose.dse



Repose the floating virtual desktop screen in front of you or on another location.

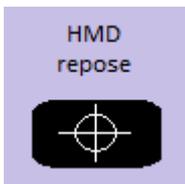
VHT_gui_switch.dse



Turn Off/On the floating virtual desktop screen.

Keyboard shortcut: Ctrl+C

VHT_hmd_repose.dse



Repose and recenter HMD in VR space.

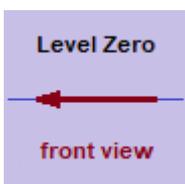
VHT_level_update (tilted view).dse



Level the scene view in VR as the Viewport camera tilt angle from that moment.

Use this to preview how the final render image will be displayed in VHT when the built-in image viewer will be used.

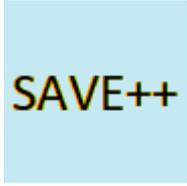
VHT_level_zero (front view).dse



Show the scene view in VR at zero degree tilt angle (default).

Miscellaneous

SAVE_incr.dse



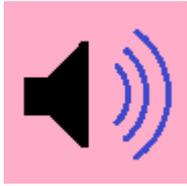
A “Save as” scene window will pop-up with a file name added using the following format:

dzsc_date-time.duf (e.g. *my_restore_point_scene_2023-10-26-17-17-47.duf*)

Use this if you want to have backup restore points.

Audio

Add_object_audio.dse



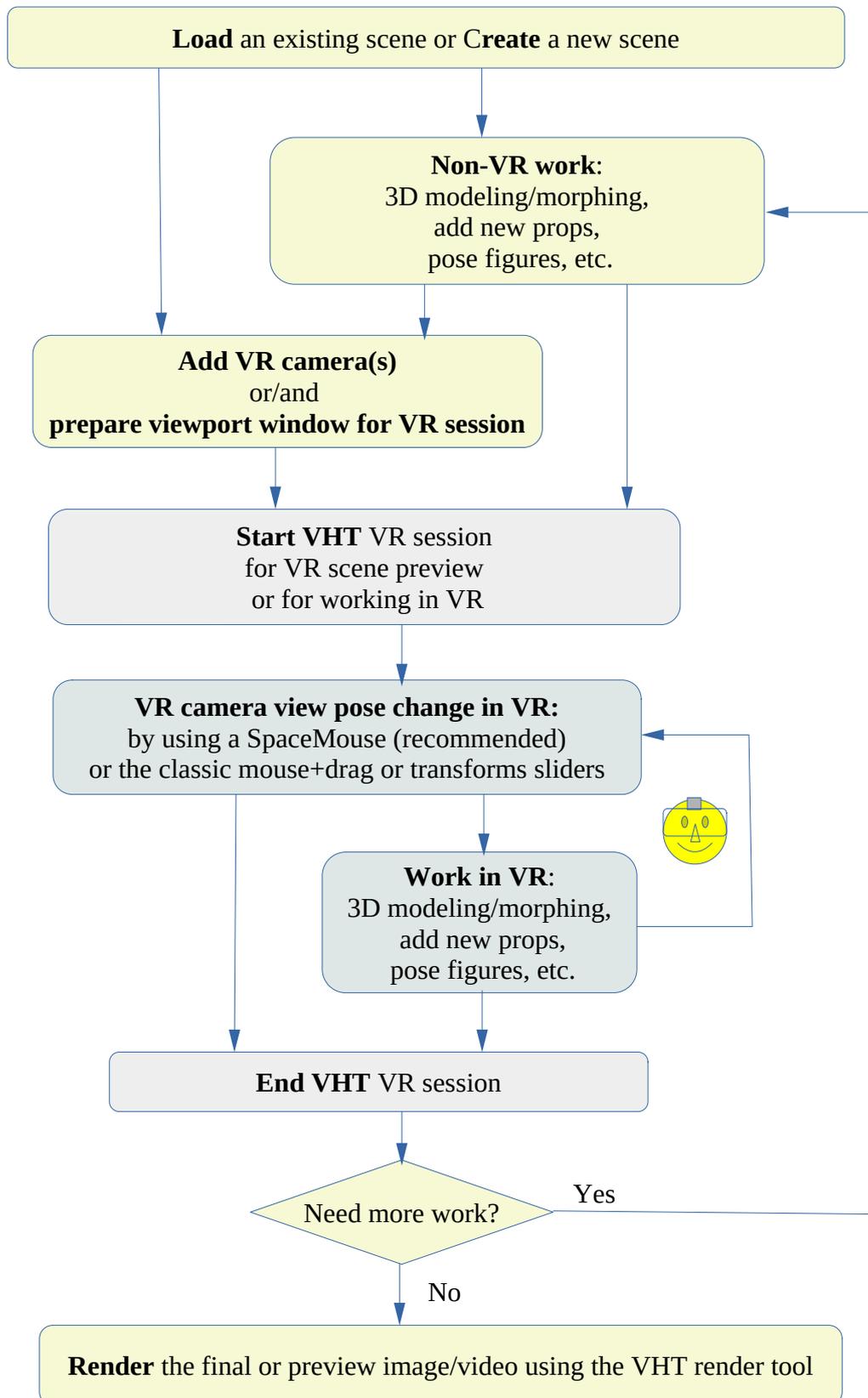
Add one Audio Object to the selected Node.

When rendering with VHT tool *Render_selected_cameras* script a companion file will be created for each rendered VR image and video.

This audiospace companion file will be used by VHT to render audio channels as virtual audio objects for binaural audio playback.

The workflow process for VR art creation with VHT tools

#1 This is the case when you have the VHT VR app (demo or paid version) installed AND you also have two displays.

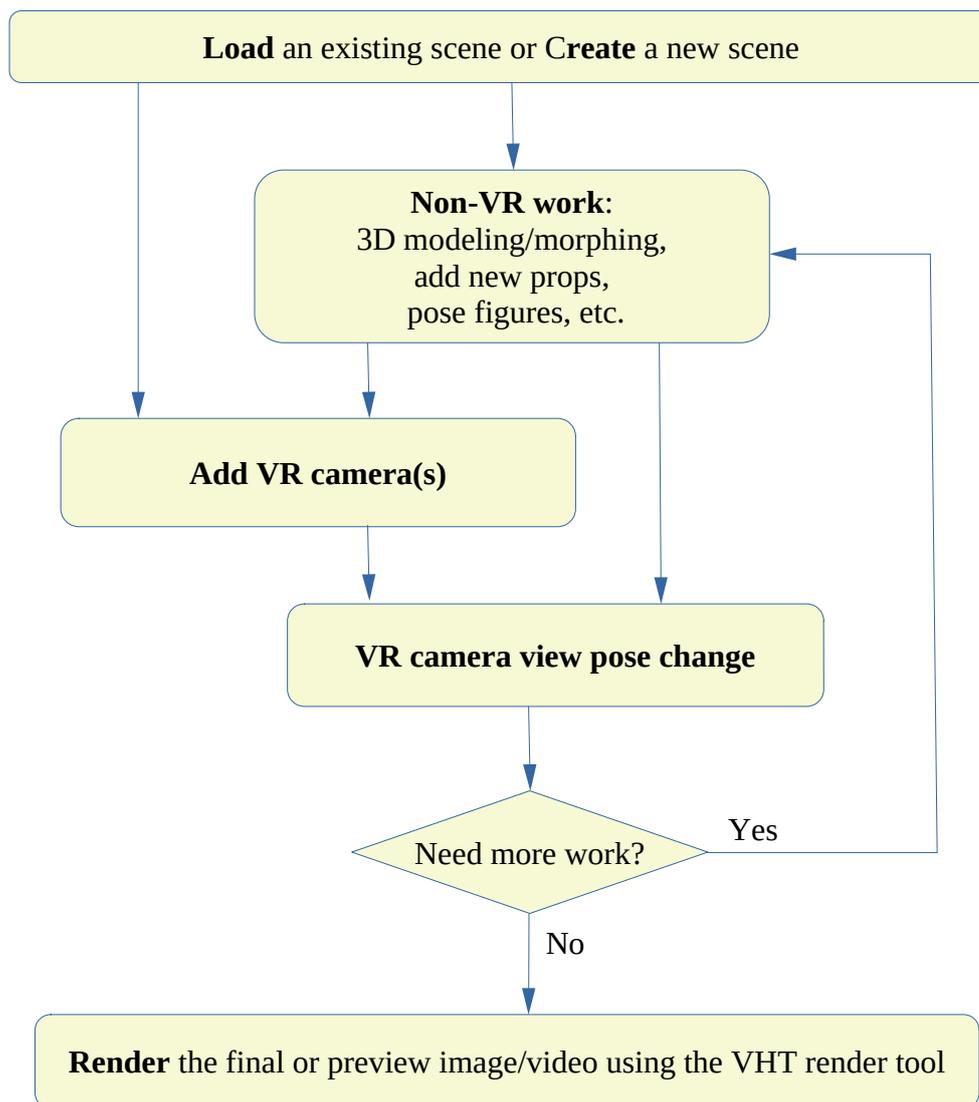


#2 The next case is when you don't have the VHT VR app (demo or paid version) installed OR you don't have two displays.

Some tools (VR cameras and render manager) can be used without VHT vr app but other tools (VR live scene preview in VR) will require VHT VR app.

When using the "VHT tools for DAZ" with VHT VR app is like using a photographic camera with a viewfinder and a tripod.

When using the "VHT tools for DAZ" without VHT VR app is like using a photographic camera and a tripod but without the viewfinder.



How to view the final rendered images or videos in VR

Use the built-in image viewer or video player from VHT (recommended) or use other VR software.

Method 1:

Right click on the image or video then “Open with” and choose VHT.

Method 2:

Use the VHT’s built-in file explorer to browse to the file location then click View or Play.

Method 3:

Use other VR software capable to correctly view the rendered stereoscopic images or videos.

Do not remove or rename the format keywords from the media file names.

For example:

Ready to Render - Barefoot Dancer R_2023-10-26-12-54-7_EQD180_FF3D_SbS_crosseye_tilt-9_.png

you can rename the “*Ready to Render - Barefoot Dancer R_2023-10-26-12-54-7*” text but keep the “*_EQD180_FF3D_SbS_crosseye_tilt-9_*” format keywords!

Working with audio objects

Audio objects are required only if you want that your CG art (image or video) to have a realistic spatial audio in VR.

These audio objects represent the location in space for a sound source that “VHT render tool” will use to create the audiospace file that VHT VR app will use to render binaural audio, delivering pinpoint audio localization in virtual space for VR art consumers.

Examples:

#1 A CG VR art image that represent a singer and a drum.

For this scene you will add one audio object to the singer mouth node and another audio object to the drum node.

Then by rendering this scene as a single stereoscopic image a companion audiospace preset file will be generated:

Singer with drum R_2023-10-26-12-54-7_EQD180_FF3D_SbS_crosseye_tilt-9_.jpg
Singer with drum R_2023-10-26-12-54-7_EQD180_FF3D_SbS_crosseye_tilt-9_.audiospace

At this stage this audiospace file can't be used by VHT because there is no audio attached to the image.

The audio file will be created by you with your preferred audio software (e.g. Audacity).

In this example you will create a dual channel audio file with one audio channel with the singer voice and another audio channel with the drums sound:

Add Singer.mp3 and drum.mp3 to Audacity then export as multichannel audio.ogg

If you add N audio objects to the scene then the final audio file must have the same number N of audio channels.

Each audio channel will represent an audio object.

Now that you have the audio file, you will need to join the image file with the audio file to create a playable media file (image+audio).

VHT will know how to play a media file like image+audio so you will not need to create a video from one single image.

In this way you will have the best image quality and the file size will be just the sum of the image file size plus the audio file size, compared with the inefficient option to create a video from a single image where the image quality will be less than the original image and the resulting media file size will be very large depending on the duration of the audio file.

To join an image with an audio file you can use the following FFmpeg command:

```
ffmpeg -i "image_name.jpg" -i "audio_name.ogg" -c copy "image_name.mkv"
```

where:

- **audio_name.ogg** is the audio file you created
- **image_name.jpg** is the rendered stereoscopic image created by VHT render tool
- **image_name.mkv** is the joined image and audio media file output

For our example it will be like:

```
ffmpeg -i "Singer with drum R_2023-10-26-12-54-7_EQD180_FF3D_SbS_crosseye_tilt-9_.jpg" -i "Singer and drum.ogg" -c copy "Singer with drum R_2023-10-26-12-54-7_EQD180_FF3D_SbS_crosseye_tilt-9_.mkv"
```

If necessary you can edit the audiospace file for channel ID to audio object matching, gain adjustment, add manually other audio objects or for direct channel playback to the headphones.

Direct channel playback to the headphones is useful if you want to add omnidirectional sounds like: echoes, hall reverb, thunder, forest ambient, rain, PoV talking etc.

#2 A CG VR art video that represent a flypast aircraft.

For this you will add one audio object to the aircraft engine node.

Render the animation by using the VHT render tool.

Now you have a video file (w/o audio) and an audiospace file with the aircraft position in space changing over time.

Then you will create a one channel audio with the aircraft engine sound.

Now that you have the audio file, you will need to join the video file with the audio file to create a playable media file (video+audio).

To join a video file (w/o audio) with an audio file you can use the following FFmpeg command:

```
ffmpeg -i "video_name.mp4" -i "audio_name.opus" -c copy "video_name.mkv"
```

where:

- **audio_name.opus** is the audio file you created
- **video_name.mp4** is the rendered stereoscopic video created by VHT render tool
- **video_name.mkv** is the joined video and audio media file output

Of course these are just basic examples but by using your imagination and your artistic skills you can create more complex scenes.

For more information about audiospace download the 'VHT Audiospace - User Guide' from VHT website!

Visual Quality and Performance Evaluation

We will evaluate the quality of stereoscopic panoramic images viewed through the HMD using the PPD property.

The type of PPD can be:

- **biological:** the human eye can have a 60 PPD in the center foveal region for 20/20 vision
- **hardware:** a consumer grade HMD can have today a 24 PPD (HP G2)
- **format:** image file resolution divided by the projection angle (e.g. for 4096x4096 at 180 degree the PPD = $4096/180 = 23$)
- **software:** this is lower or equal than the format PPD and depends on the compression level, render samples and denoise level

The quality of an image in VR will be defined by the software PPD and limited by format and hardware PPD.

The perceived PPD (what you actually see in VR) will be the minimum PPD of the above types.

For the following use cases (art creation and consuming art) we will use a HP G2 and NVIDIA RTX 4090 GPU.

1st Art preview in VR

This is the case when an artist use VHT tools for DAZ to create VR art in VR for VR users.

He has a 2nd display with a resolution of 2560x1440 @ 60Hz.

When using the OpenGL raster render engine for EQD180 or REL9mm the software PPD will be always equal with the format PPD of $2560/2/127 = 10$ (close to Valve Index HMD PPD of 13) at a variable and lower than 60 fps depending on the complexity of the scene (usually will be at 60 fps for most cases).

When using NVIDIA Iray render engine with 32 samples for Viewport and denoiser for EQD180 the software PPD will be lower than theoretical format PPD of $2560/2/180 = 7$.

For each render cycle the software PPD will increase from zero to 7 PPD in 32 steps at a rate of 5 fps. Depending on the scene complexity the ray traced image will be updated at a rate of <5 fps (this fps is OK for a static scene turn based live preview ray-traced render in VR).

2nd Rendering the final VR art

This is the case when an artist is happy with the scene he created and want to share his art with others. He will use the VHT tools for DAZ to render the VR art.

You should expect double render duration for a stereoscopic image due to the format nature of this kind of image (need one image for left and another for right eye) for a given resolution.

If you choose the best render setting: samples > 500, Rendered Converged Ratio > 90 % and png/tif you should expect the software and format PPD for:

- EQD180 at 4096x4096/eye resolution: $4096/180 = 23$ PPD
- REL9mm at 4096x4096/eye resolution: $4096/127 = 32$ PPD
- VR360 at 8192x4096/eye resolution: $8192/360 = 23$ PPD

Render times will range from minutes to few hours per one final stereoscopic (left+right) image depending on scene complexity, format and software target PPD, GPU, etc.

3rd Consuming the VR art

This is the case when a VR user will admire your VR art.

For a rendered EQD180 image he can enjoy a $4096/180 = 23$ PPD image details which is a perfect match to his HP G2 VR HMD of 24 PPD.

He will be very impressed by this 4K per eye 3D VR panoramic image.

Demo samples

For rendered images and videos with or without audio objects go to <https://virtualhometheater.com/>

Acronyms

- VHT – Virtual Home Theater
- DS – Daz Studio
- DAZ – Daz Studio
- HMD – Head Mounted Display (VR device)
- CGI – Computer Generated Image
- CG – Computer Graphics
- IPD – Interpupillary distance (distance between two cameras or eyes)
- PPD – pixels (details) per angle degree (hardware or visual)
- SbS – Side-by-Syde (screens layout format for stereoscopic images)
- REL – Rectilinear (photographic projection/mapping)
- EQD – Equidistant (photographic projection/mapping)
- FPV – First Person View (the view point of a character from the scene)
- AO – Audio Object
- FoV – Field of View
- GUI – Graphical User Interface
- UI – User Interface
- PC – personal computer
- IPC – inter-process communication
- PoV – point of view

Links

Virtual Home Theater:

- <https://virtualhometheater.com/>
- <https://store.steampowered.com/app/989060/>
- <https://www.youtube.com/@VHTtools>
- <https://www.youtube.com/@virtualhometheater3887>
- <https://youtu.be/r4DJWKSygPQ>

Daz Studio:

<https://www.daz3d.com/>

<https://raytracing-docs.nvidia.com/iray/manual/index.html#preface#>

Spherical Stereoscopic Panoramas:

<https://paulbourke.net/papers/vsmm2006/>

<https://ffmpeg.org/>

<https://www.audacityteam.org/>

Revision History

- v1.0 – Feb 17, 2024: First release