

# VIOSO Unity Plugin

---

## Presentation

The plugin lets you integrate your VIOSO display calibration directly into your Unity Application.

## Manual

### In VIOSO software

After a successful calibration of your displays, export your mapping in the VWF format:

From the menu bar, go to **File > Export Mapping**

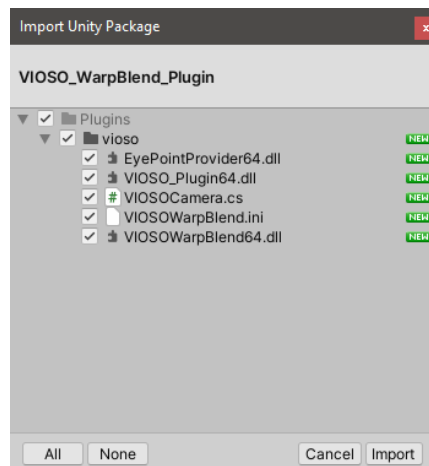
- Select the **VWF** format.
- Uncheck the “**use settings from file**” and check the “**3D**” box if applicable.
- Choose a file name and path.
- Click on **Export**.

---

### In Unity

1. Open your Unity Project.
2. Add “**VIOSO\_Unity\_Plugin.unitypackage**” to your project.

Assets > Import Package > Custom Package

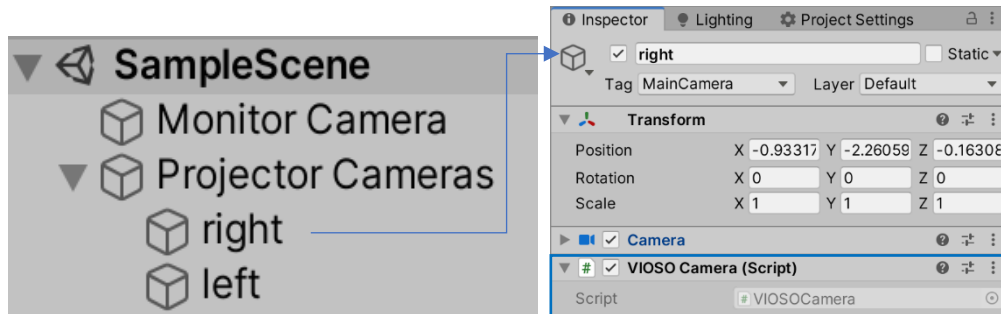


Click on “**Import**”

3. Navigate to the plugin directory: *Assets\Plugins\vioso*  
And add your mapping file(s) (**.VWF**) to that folder.

4. Add “**VIOSOcamera.cs**” script to each of your Camera Objects.  
(Assets\Plugins\vioso\VIOSOcamera.cs)

(!) The Camera for the main monitor (Display 1) should not contain the script, as it’s usually not part of the calibration.



5. **VIOSOWarpBlend.ini** Configuration:

- Open Assets\Plugins\vioso\VIOSOWarpBlend.ini
- Add the **name(s) of your .VWF** and the **parameters for your cameras** as shown below:

```
1st case: 1 VWF file, Multiple Displays

[default]
LogLevel=2

;eyePointProvider=EyePointProvider64
eyePointProviderParam=sinewave

base=[1000, 0, 0, 0; 0, 1000, 0, 0;
0, 0, -1000, 0; 0, 0, 0, 1]
near=0.15000
far=1000.00000
screen=2.00000
autoViewC=1.00000;
bAutoView=1;
bTurnWithView=1

calibFile=Panadome3D.vwf

[left]
calibIndex=0
[center]
calibIndex=1
[right]
calibIndex=2
```

```
2nd Case: Multiple VWF files
(Applicable to Multi-client)

[default]
LogLevel=2

;eyePointProvider=EyePointProvider64
eyePointProviderParam=sinewave

base=[1000, 0, 0, 0; 0, 1000, 0, 0;
0, 0, -1000, 0; 0, 0, 0, 1]
near=0.15000
far=1000.00000
screen=2.00000
autoViewC=1.00000;
bAutoView=1;
bTurnWithView=1

calibFile=left.vwf
[left]

calibFile=center.vwf
[center]

calibFile=right.vwf
[right]
```

- **calibFile**= Name of .VWF file(s).
- **[name]** : Should be the name of the **Camera GameObject in Unity**.
- **CalibIndex** : Assign the Index according to your VIOSO calibration.

*Optionally, you can:*

Add your **Pivot Point** coordinates by editing the highlighted **X,Y,Z** parameters of the **base** matrix as shown below:

**base** = [1000, 0, 0, 0; 0, 1000, 0, 0; 0, 0, -1000, 0; **X, Y, Z**, 1]

(X,Y,Z)=(0,0,0) being the centre of the projection screen.

6. The Integration is now completed. You can preview it by running the player and switching between displays (see note n°2). If everything looks good, **Build** your project.
7. Copy the **.VWF** files and **"VIOSOWarpBlend.ini"** from Unity **Assets/plugin/vioso** to **[your\_build\_path..]/\_data/plugin** next to ViosoWarpBlend.dll.
8. Launch your application and enjoy the integrated calibration.

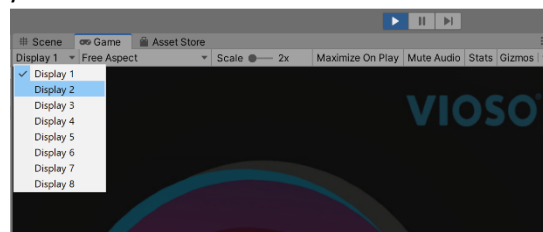
## Notes & Common bug fixes:

- Make sure you **activate multi-display** support in your Unity Application.

Unity's default display mode is one monitor only. When you run your application, you need use `Display.Activate()` to explicitly activate additional displays. Follow the Unity Manual:

<https://docs.unity3d.com/Manual/MultiDisplay.html>

- The multi-display feature of Unity only runs in the built standalone application. However, you can **preview your displays** in the Editor by running the project (**Play** button) then navigating through displays in the **Game tab** to see the different calibrated outputs.



- Make sure you **only have one audio listener in the scene**. Therefore, Remove this component from all your cameras except one:



- **Dynamic Eyepoint**

Enable Dynamic Eyepoint in **VIOSOWarpBlend.ini** by setting **eyePointProviderParam** to listen to a specific port using UDP protocol (i.e = listen 999) or for a simple test you can use **=sinewave** to undulate all axes and angles in a loop. See VIOSOWarpBlend documentation for further details.

- You can manually add the following parameters to your cameras in **VIOSOWarpBlend.ini**:

**dir**= [x,y,z] : Direction.

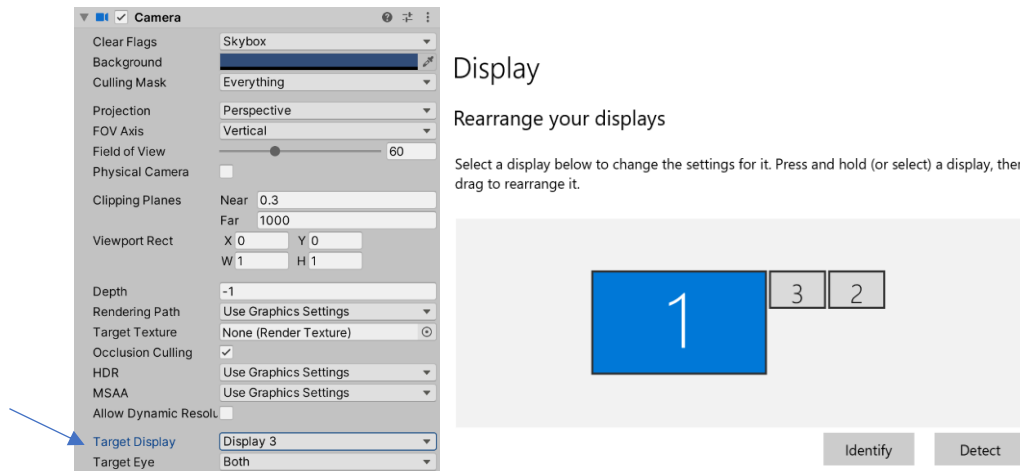
**fov**= [x,y,z,w] : Field of view.

They will be **recalculated automatically** and overwritten if **bAutoView = 1**.

- **BUG: Cameras not displaying in the correct order:**

1<sup>st</sup> fix: Camera Target Display

- Select a camera.
- From the Inspector window go to Camera > Target Display
- Select the corresponding display according to the index from the Windows Display Settings.



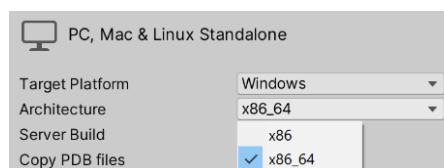
2<sup>nd</sup> fix: Camera Calibration Index

Open the **.VWF** file and verify that the calibration **index** corresponds to the one you assigned in the **VIOSOWarpBlend.ini** file.



- Make sure you are building your project for the **x86\_64** Architecture.

File > Build Settings



- Please check **VIOSO\_Plugin64.log** in the build directory's plugin folder to examine the status and errors of the integration.

## Support

Should there be any questions, problems or errors, please reach us through our online support platform:

[www.vioso.com/support](http://www.vioso.com/support)

Our qualified team members will assist you with all testing and production issues. Our services also include on-site project support and training.

### **VIOSO GmbH**

Ronsdorferstr. 77a D-40233 Düsseldorf

Tel. +49 211 544 75 33 – 0

Fax: +49 211 544 75 33 – 33

E-mail: [info@vioso.com](mailto:info@vioso.com)

Internet: [www.vioso.com](http://www.vioso.com)