



miniSim™

IOWA

National Advanced Driving Simulator

## NADS miniSim Viewport and Desktop Settings

Andrew Veit  
andrew-veit@uiowa.edu  
(319 )594-0751

November 30, 2022

### Contents

- Introduction ..... 2
- Desktop Layout ..... 3
- Display Connections and Settings ..... 5
  - GeForce/Matrox..... 5
  - Matrox Power Desk..... 7
  - Typical Resolutions ..... 7
  - Channel Configuration File Edits ..... 8
  - Channel ID Determination – Explicit Method ..... 8
  - Quadro/Mosaic ..... 10
  - Channel Configuration File Edits ..... 13
  - Channel ID Determination – Explicit Method ..... 13
- Troubleshooting ..... 14
- The Channel Configuration File (.ccf) ..... 15
  - Channel Blocks ..... 18
  - Viewport Blocks ..... 19
- Viewport Settings ..... 22
  - Viewport Background: ..... 22
  - miniSim Viewports for Three 48” Displays ..... 25
  - Non-Standard Viewport Setup Example ..... 26
  - Measurements ..... 26
  - Calculation ..... 27
- Display Performance ..... 30

## Introduction

The miniSim rendering system can generate discrete rendering windows called **channels**. Each channel can be divided into **viewports**. Each channel is typically a discrete video output, i.e. a HDMI, DisplayPort, or DVI jack on a graphics card. A viewport defines the viewing angle and field of view (FOV) for a display or projector. A channel may have many viewports. When larger displays are used, the viewport is configured to represent the virtual world at 1:1 scale. With small displays, this is not practical, and the virtual world is typically rendered at a smaller scale in order to obtain enough field of view to effectively 'drive' the simulator.



The desktop configuration for all the miniSim displays is very important. The desktop display layout and the Channel Configuration File (.ccf) work together to display images on the correct screen at the correct scale.

## Desktop Layout

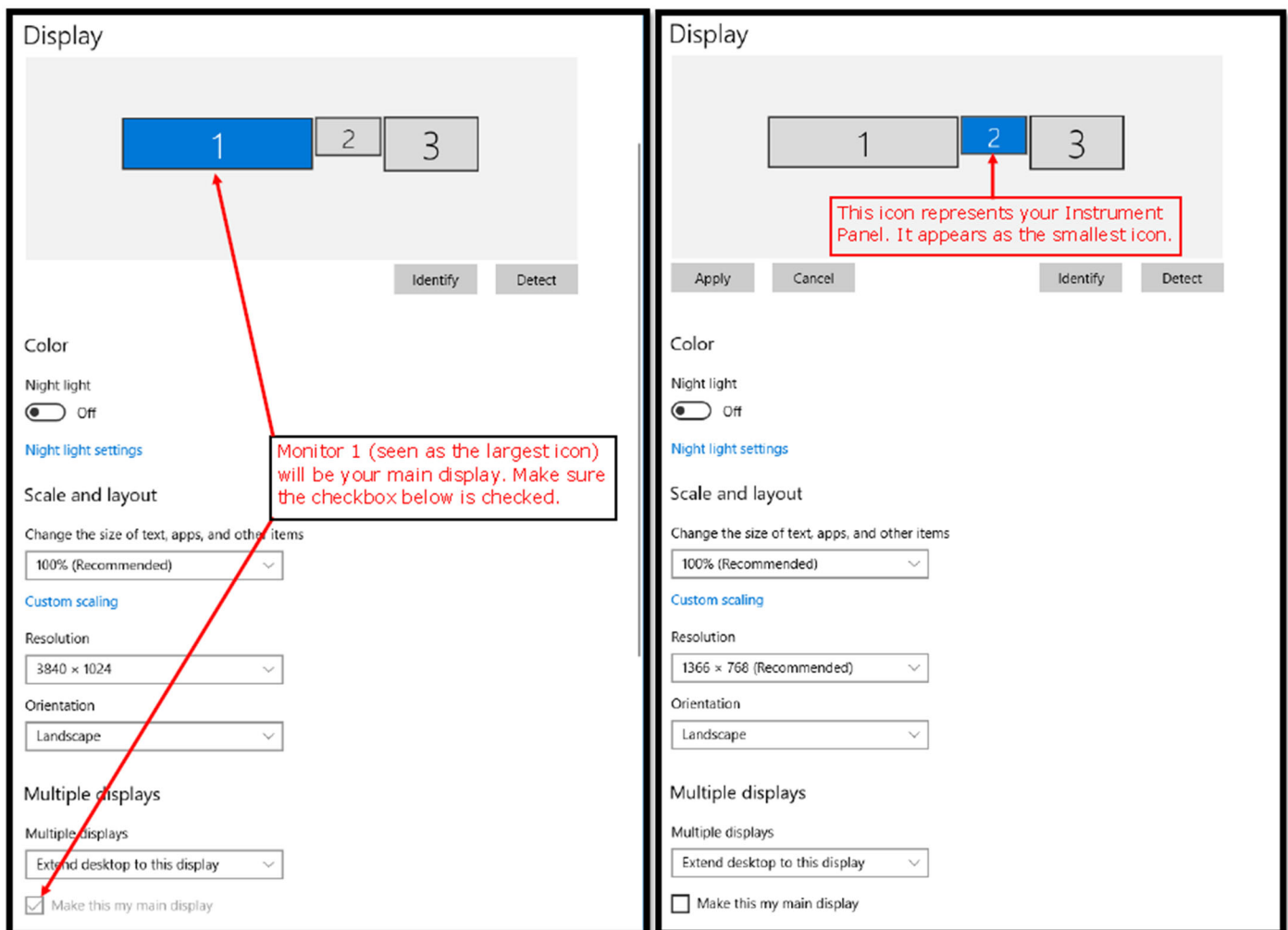
The miniSim desktop layout is critical for proper operation. On all 3-screen miniSims, the front 3 displays are combined into a single display (channel). This is accomplished with either a hardware device called a Matrox, or using an NVIDIA feature called Mosaic (Quadro cards only).

In either case, the Matrox or Mosaic will appear to the operating system as a **single** display, with the pixel count and aspect ratio of three displays arranged in a horizontal line. The instrument panel and operator displays will each appear as single displays. When configured correctly, the desktop will look like this (right-click on the desktop and choose 'Display Settings').

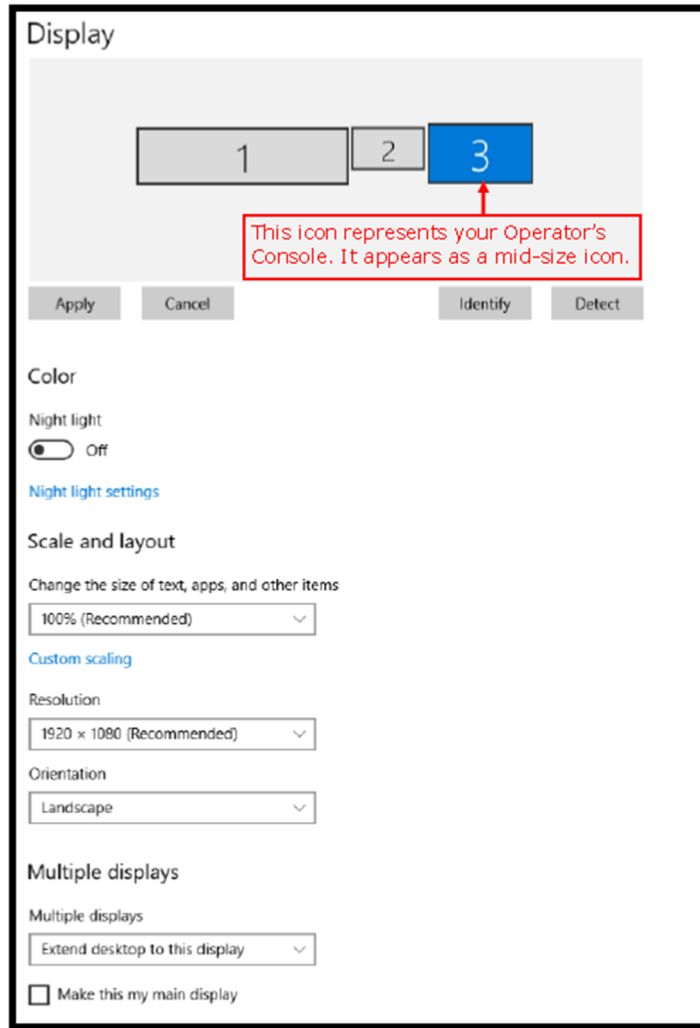
It is critical that the following are true:

- Displays must be arranged from left to right: **Matrox/Mosaic, Instrument Panel, and Operator**. These are moved by dragging with mouse.
- The top edges of the icons are aligned (use mouse or arrow keys)
- The Matrox or Mosaic display **must** be selected as the 'main' or 'primary' display in the 'Display Settings' or 'Display Resolution' menu of the Windows operating system.

To access this window, right-click on the desktop somewhere and select 'Display Settings' (in Windows 10) or 'Display Resolution' (in Windows 7) from the menu. You will see the following window. Click on each display icon to see its properties:

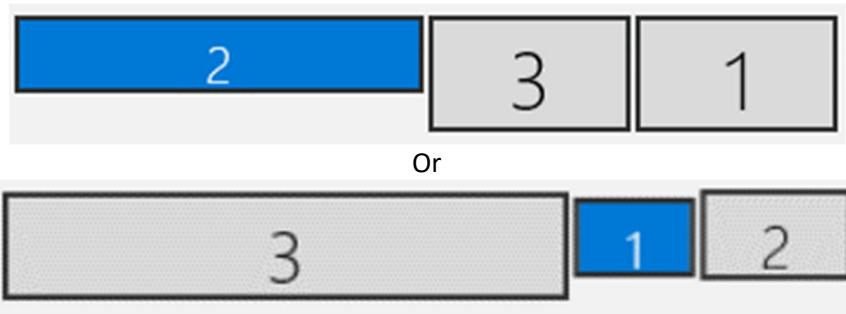


Right click on the desktop and select 'Display Settings' or 'Display Resolution' to access this menu



Right click on the desktop and select 'Display Settings' or 'Display Resolution' to access this menu

Note that the Operating System and Video Driver assign the display numbers. It may be any combination, such as:



The numbering, **does**, however affect a **setting** in the **channel configuration file**, described later:

```

Channel 0 0 1 (Shear x2) (Twist x3) (Extends)
0.000000 0.000000
0.000000 0.000000 0.000000
0.000000 0.000000 1.000000 1.000000
6

```

## Display Connections and Settings

### GeForce/Matrox

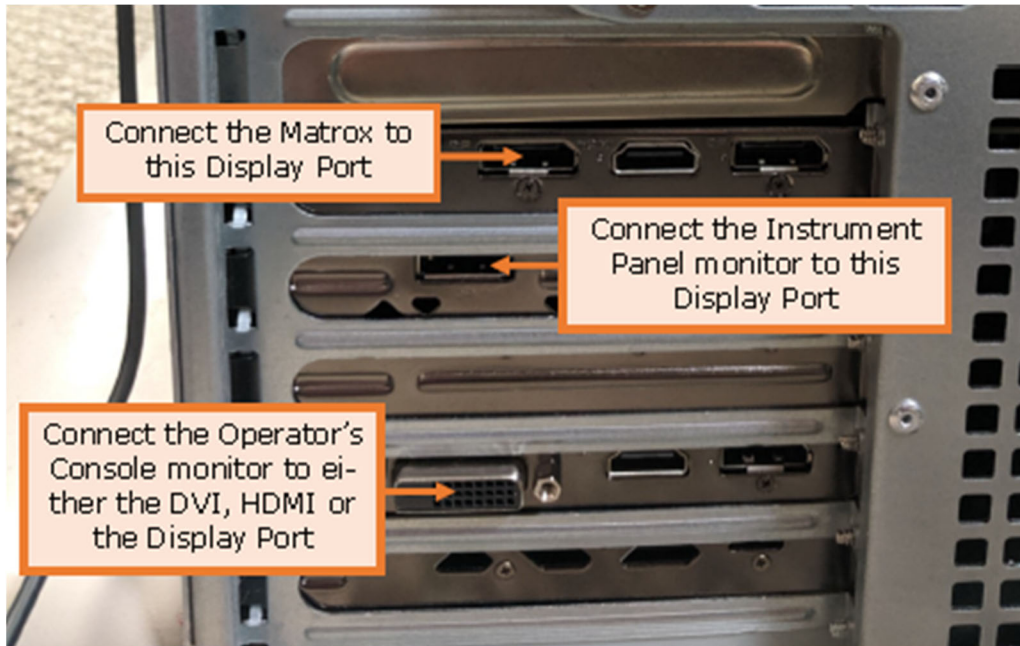
All miniSim systems have two video cards. One card is a high performance model, such as a GeForce 80 or 70 series (GTX 880, 980, 3080, 3070 etc), and a lower-performance model (GTX 1650). The Matrox and Instrument Panel displays are connected to the high performance card, and the operator display is connected to the lower performance model. The [Matrox](#) is a hardware module that combines three video outputs into a single, very wide display.



**Matrox device with 1 Display Port Input, and 3 DVI outputs.**

The Matrox, depending on the model may have DisplayPort or DVI inputs and outputs. The [Matrox](#) can only support (up to) the following resolution at **60Hz**:

- Front: Left, Center, Right displays/projectors: 3 x 1600 x 900 (3 x 1280 x 1024 typ.)

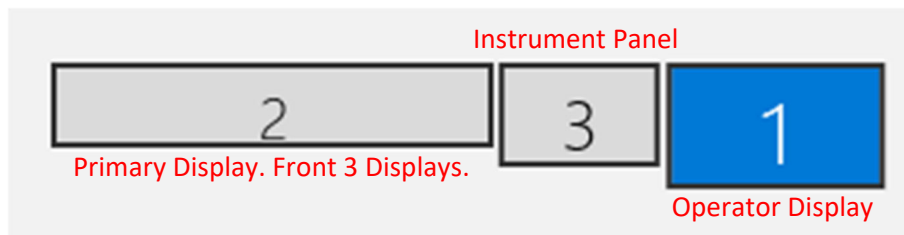


Monitor connection locations

To access this window, right-click on the desktop and select 'NVIDIA Control Panel' from the listed options. Then choose 'Set up Multiple Displays' from the list on the left of the NVIDIA window.



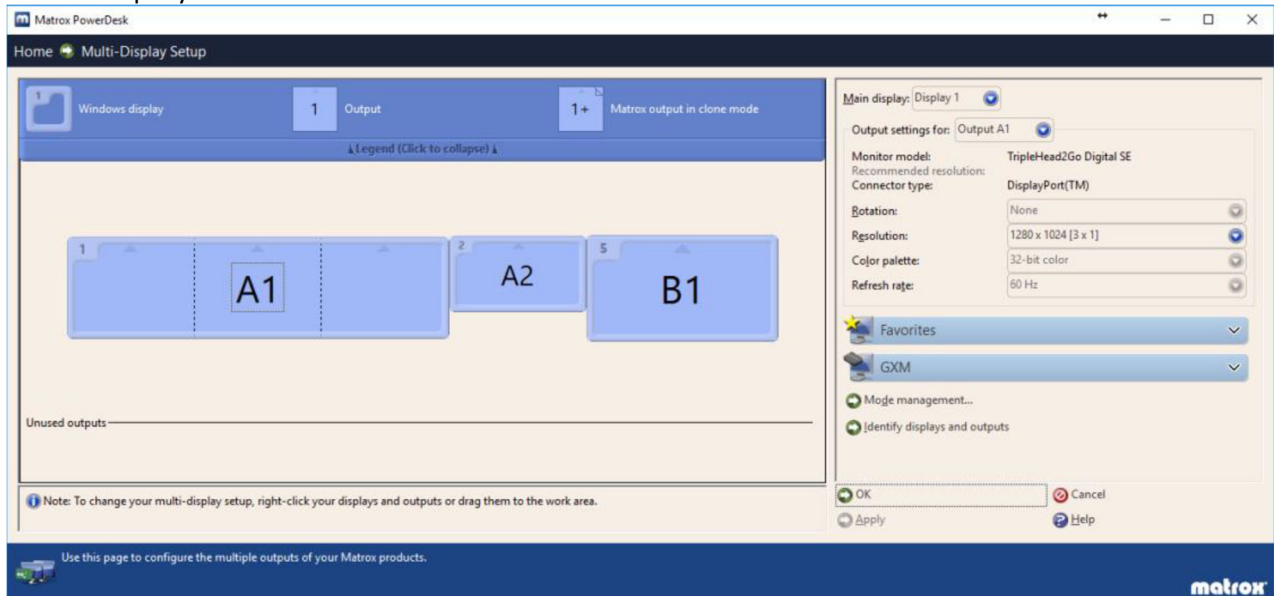
Monitor connections as viewed in NVIDIA 'Set Up Multiple Displays' window.



Desktop as viewed in Windows Display Settings (these do match the NVIDIA numbering in this case)

## Matrox Power Desk

You may notice an application on your miniSim called **Matrox PowerDesk**. If you open it, you can view the desktop layout here as well. It will look like this:



The desktop configuration and resolution settings shown here will match that in the NVIDIA driver and Windows. The identifiers used, however, are unique for PowerDesk. It's good to confirm that the desktop layout in PowerDesk matches that in Windows and NVIDIA. Sometimes after you change the desktop configuration, PowerDesk will pop up a message asking to reboot Windows. Always answer 'yes'. It may take one or more reboots until all the display settings match across Windows, PowerDesk, and the NVIDIA driver.

PowerDesk can be set to control what display application windows appear on, and where Windows system windows appear, however, we **do not recommend** the use of this feature, as it will have unpredictable effects on miniSim operation.

## Typical Resolutions

This hardware supports the following number of video channels and resolutions at **60Hz**:

- Front: Left, Center, Right displays/projectors: 3 x 1600 x 900 (3 x 1280 x 1024 typ.)
- Instrument Panel: 16:9 LCD, up to 1920x1080, (1280x800, 1600x900 typical)
- One Auxiliary Video Output: up to 1920x1080 (1280x800, 1600x900 typical)
- Operator Display: 1920x1080

If higher resolution is needed, then Quadro cards must be used.

## Channel Configuration File Edits

The channel configuration file (.ccf) channel headers look like this. The highlighted number can be 0, 1, or 2. It is set to the number of the main display minus 1. So in this case it is 2-1, or 1.

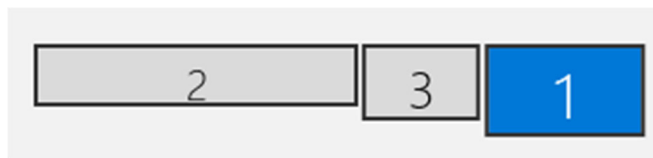
You can always experiment with 0, 1, and 2 if needed, closing and launching the miniSim each time to make the change take effect.

```
Channel 0 0 1 (Shear x2) (Twist x3) (Extends x4)
0.000000 0.000000
0.000000 0.000000 0.000000
0.000000 0.000000 1.000000 1.000000
6
.
.
.
Channel 1 0 1 (Shear x2) (Twist x3) (Extends x4)
0.000000 0.000000
0.000000 0.000000 0.000000
1.000000 0.000000 2.000000 1.000000
0 IP
```

## Channel ID Determination – Explicit Method

An explicit method exists to identify the correct setting for the critical third digit in line 1 of the Channel ID block. This method uses the Open Scene Graph library in miniSim to render a model on a specific screen to verify the correct value to use.

Using this layout as an example:



Based on the above process, we think the third digit in line 1 of the Channel ID block, should be 1. We can test hypothesis by following these steps:

1. Open a *Command Prompt* (search for 'CMD' in the search window by the start menu)
2. At the prompt, type the following. After each line, hit 'Enter'.

```
cd C:\NadsMiniSim_2.3\bin.x64
SetupEnv.cmd
cd ..\data\Visuals\Models
osgviewer --screen 1 drum.ive
```
3. The image of the 'Drum' model will appear in the center of the front screens. This is the correct location for the first of the miniSim rendering windows to appear, so 1 is the correct value to use.





4. Hit the 'Esc' key to exit. You may need to click on the front window first to bring the windows focus there before hitting the 'Esc' key.
5. If the Drum is not on the front screen, try 0 or 2 etc. to find the correct screen.

Note: The messages in the command window look like this. The commands you type in are in bold for clarity:

```
Microsoft Windows [Version 10.0.19042.867]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\minisim>cd C:\NadsMiniSim_2.3\bin.x64

C:\NadsMiniSim_2.3\bin.x64>SetupEnv.cmd

~a bunch of messages appear~

C:\NadsMiniSim_2.3\bin.x64>cd ..\data\Visuals\Models

C:\NadsMiniSim_2.3\data\Visuals\Models>osgviewer --screen 1 drum.ive
library to be loaded is osgPlugins-3.1.0/osgdb_ive.dll
library to be loaded is osgPlugins-3.1.0/osgdb_dds.dll
```

The Drum is displayed on screen **1**. If it is not on the front screen, try another number. Click on the front screen and hit 'Esc' to close the OSG window.

**Note:** Useful commands are:

cd	Change Directory
cd ..	move up one directly level
dir	list the contents of the current directory
'TAB' key	Will auto-complete a folder or file name if it exists in the current directory. Start typing the desired directory, for example, type 'nads' and hit the 'TAB' and it will suggest a name that matches. Keep hitting TAB until NadsMiniSim_2.3 appears.

It is also handy to follow along in the File Explorer window, as its better for visualizing the directory structure.

References:

[https://en.wikipedia.org/wiki/List\\_of\\_DOS\\_commands](https://en.wikipedia.org/wiki/List_of_DOS_commands)

[https://en.wikipedia.org/wiki/Cd\\_\(command\)](https://en.wikipedia.org/wiki/Cd_(command))

## Quadro/Mosaic

Quadro cards allow the system to drive the front 3 displays at full HD resolution (1920x1080) at 60 Hz. This is typically needed with larger displays and projectors to obtain good image quality.

Similar to the GeForce/Matrox settings above, these systems have a high-performance (e.g. Quadro P6000) and lower-performance card (e.g. Quadro P4000). Things are a little different here, as ultimately the monitors we want to combine with Mosaic need to be all the same resolution and need to be connected to the high-performance card. In this example, the front 3 channels are on the Quadro RTX 6000, and the Instrument Panel and Operator Display are on the Quadro RTX 4000:



Display Connections for a Quadro based system with 3 front displays

Mosaic Displays- 1
Modify
Disable

Topology	1 x 3
Refresh Rate	60Hz
Resolution per display	1920 x 1080
Total resolution	5760 x 1080

Sync Capability Information

GPU/ Displays	Sync capability
Quadro RTX 4000	
1. HP 2009	
2. WXGA TFT LCD	
Quadro RTX 6000	
3. SONY TV	
4. SONY TV	
5. SONY TV	

Three Sony displays are combined into a single display using the NVIDIA Mosaic function.

Notice the 3 Sony TVs (3,4,5) are on the RTX 6000 card.

When combined in Mosaic they become display 5 in the NVIDIA application

The Operator display and Instrument Panel are on the RTX 4000 card (1,2)

**1. Select the displays you want to use.**

Quadro RTX 4000

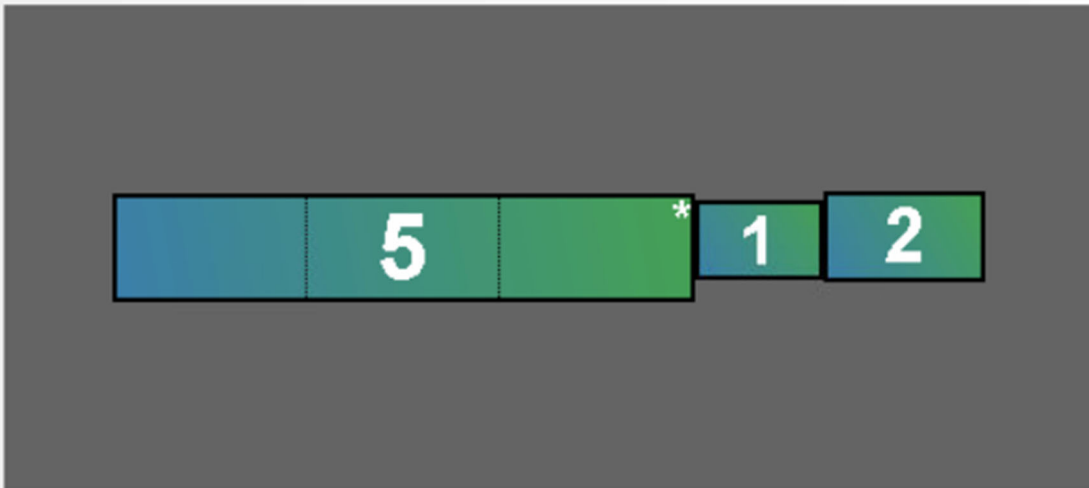
- HP 2009 **2** Operator Display
- Toshiba Matsushita Display Technology ... **1** Instrument Panel

Quadro RTX 6000

- SONY TV (2 of 3) **5** Primary Display. Front 3 Displays in Mosaic

[My display is not shown...](#)

**2. Drag the icons to match your display configuration.**



[Mosaic options](#)

Sample desktop in NVIDIA once the Mosaic is engaged. Again, numbering may vary.



The same desktop in Windows Display Settings once the Mosaic is engaged.

Note the Number 5 in NVIDIA is displayed as 3 in Windows.

Channel start number in the CCF in this case is 2.

## Channel Configuration File Edits

The channel configuration file (.ccf) channel headers look like this. The highlighted number can be 0, 1, or 2. It is set to the number of the main display minus 1. So in this case it is 3-1, or 2.

You can always experiment with 0, 1, and 2 if needed, closing and launching the miniSim each time to make the change take effect.

```
Channel 0 0 2 (Shear x2) (Twist x3) (Extends x4)
0.000000 0.000000
0.000000 0.000000 0.000000
0.000000 0.000000 1.000000 1.000000
6
.
.
.
Channel 1 0 2 (Shear x2) (Twist x3) (Extends x4)
0.000000 0.000000
0.000000 0.000000 0.000000
1.000000 0.000000 2.000000 1.000000
0 IP
```

## Channel ID Determination – Explicit Method

You can also use the same alternative method described in the Matrox section above using the command prompt.

## Troubleshooting

The most common problem is when the miniSim channels appear on the wrong displays. What is happening?

The fundamental answer is that:

- a. The desktop layout is incorrect
- b. The relationship between the display layout and the settings in the appropriate Channel Configuration File (.ccf) **do not match**.

To resolve this, check the following in this order:

1. Displays are arranged from left to right: **Matrox/Mosaic, Instrument Panel, Operator**. (use mouse or arrow keys)
2. The Matrox/Mosaic display **must** be designated as the 'main' or 'primary' in Windows. The task bar may move to this display when this is set, so drag the task bar and icons to the operator display
3. **Do not** use the Matrox PowerDesk program to adjust the default locations for pop-up and application windows. This will cause many, many problems.
4. The **top** edges of the display icons are aligned (use mouse or arrow keys)
5. Reboot to make sure the configuration is stable.
6. Check video and USB connections. If they are loose, it may cause windows to re-number the displays, which can make things move around unpredictably.
7. Open the **correct** Channel Configuration File (.ccf) in the \bin.x64 directory. Note there are several .ccf files in every system, and the 'correct' one is defined by the system settings in the 'go.cmd' file used to start miniSim. See next section. The third number in the Channel header(s) should be equal to the number of the main display as reported by Windows in 'Display Settings' minus 1. Only 0, 1, 2 are generally valid for standard miniSim configurations (2 Channel, 3 displays).

```
Channel 0 0 2 (Shear x2) (Twist x3) (Extends x4)
0.000000 0.000000
0.000000 0.000000 0.000000
0.000000 0.000000 1.000000 1.000000
6
.
.
Channel 1 0 2 (Shear x2) (Twist x3) (Extends x4)
0.000000 0.000000
0.000000 0.000000 0.000000
1.000000 0.000000 2.000000 1.000000
0 IP
```

## The Channel Configuration File (.ccf)

The minisim is launched by running the correct **go.cmd** file in the **NadsMiniSim\_x.x.x\bin.x64** folder. Typically, there have been shortcuts created on the miniSim desktop that are used to run the appropriate 'go' scripts. There are often several go-scripts, this is usually to provide a way to launch the miniSim with or without various options (Video Capture, Advanced Automation, etc).

If you right-click on the shortcut icon and choose 'edit' you can edit the go.cmd script. A typical one looks like this:

```
set MININADSCOMMMODE=PSEXEC
set DYNA=car
set NUMCHN=2
set INSTRUMENTS=on
set FORMAT=wide
set INSETVIEW=off
set SIDEMIRRORS=on
set MININADSDAQ=on
```

Based on these settings, the miniSim will read the following channel configuration file (.ccf) and use its contents for setting up the rendering channels and viewports. In this case, the miniSim will open this file:

**\bin.x64\mininads\_2ch.wide.sidemirrors.inst.ccf**

This is the correct file to edit for this configuration. **If you make a change and see no effect, you may be editing the wrong file!**

When the miniSim runs, it copies the .ccf file that pertains to the options selected in the go.cmd, and copies it to **mininads\_1ch.ccf**, which is the one that is loaded when miniSim is launched.

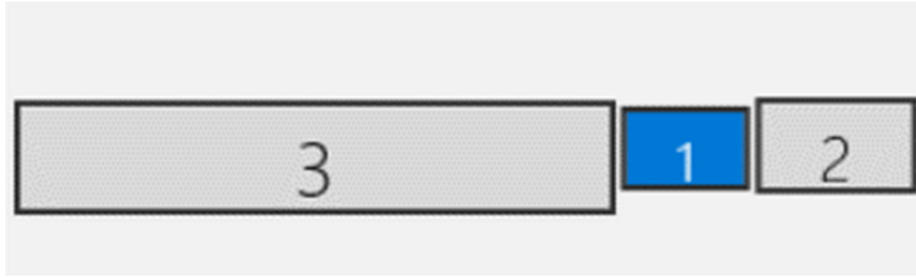
This means that the **mininads\_1ch.ccf** has the same 'Last Modified Date' as the file you want to edit.

If you list the contents of the \bin.X64 folder by date, you will see something like this:

test_all.scl	3/3/2021 11:38 AM	SCL File	287 KB
TTAtest.txt	3/1/2021 6:29 PM	Text Document	1 KB
mininads_1ch.ccf	2/26/2021 1:23 PM	CCF File	2 KB
mininads_2ch.wide.sidemirrors.inst.ccf	2/26/2021 1:23 PM	CCF File	2 KB
miniSimMotion.exe	2/24/2021 3:55 PM	Application	125 KB
dbxLive32.dll	2/24/2021 3:55 PM	Application exten...	4,564 KB

In this case, you will see that the **mininads\_2ch.wide.sidemirrors.inst.ccf** is the file you want to edit, as it has the same last modified date as **mininads\_1ch.ccf**.

For a desktop layout like this from Windows in 'Display Settings'



This is a **Two-Channel** system, meaning that the front three displays on the Matrox/Mosaic are Channel 0, and the Instrument Panel display is Channel 1.

The contents of [mininads\\_2ch.wide.sidemirrors.inst.ccf](#) look like:



Channel 0 0 2 (Shear x2) (Twist x3) (Extends x4)  
 0.000000 0.000000  
 0.000000 0.000000 0.000000  
 0.000000 0.000000 1.000000 1.000000  
 6  
 Viewport 0 (AOV x1) (Clip x2) (Extends x4) (Offs x6) (Flip x2)  
 46.60  
 0.300000 35000.000000  
 -1.000000 -1.000000 -0.333333 1.000000  
 0.000000 -1.10000 0.000000 47.870000 0.000000  
 0 0 LV  
 Viewport 1 (AOV x1) (Clip x2) (Extends x4) (Offs x6) (Flip x2)  
 46.84  
 0.300000 35000.000000  
 -0.333333 -1.000000 0.333333 1.000000  
 0.000000 -1.100000 0.000000 0.000000 0.000000  
 0 0 CV  
 Viewport 2 (AOV x1) (Clip x2) (Extends x4) (Offs x6) (Flip x2)  
 46.60  
 0.300000 35000.000000  
 0.333333 -1.000000 1.000000 1.000000  
 0.000000 -1.100000 0.000000 -47.870000 0.000000  
 0 0 RV  
 Viewport 3 (AOV x1) (Clip x2) (Extends x4) (Offs x6) (Flip x2)  
 42.000000  
 0.700000 100000.000000  
 0.346667 0.555000 0.611667 0.865000  
 0.000000 0.000000 0.000000 0.000000 0.000000  
 1 0 CM  
 Viewport 4 (AOV x1) (Clip x2) (Extends x4) (Offs x6) (Flip x2)  
 27.000000  
 0.700000 100000.000000  
 -0.632000 -0.995000 -0.438000 -0.449000  
 0.000000 0.000000 0.000000 0.000000 0.000000  
 1 0 LM  
 Viewport 5 (AOV x1) (Clip x2) (Extends x4) (Offs x6) (Flip x2)  
 25.000000  
 1.100000 100000.000000  
 0.894333 -0.714000 0.984000 -0.346000  
 0.000000 0.000000 0.000000 0.000000 0.000000  
 1 0 RM  
 Channel 1 0 2 (Shear x2) (Twist x3) (Extends x4)  
 0.000000 0.000000  
 0.000000 0.000000 0.000000  
 1.000000 0.000000 2.000000 1.000000  
 0 IP

## Channel Blocks

This example is a two Channel system, so there are two Channel headers, 0 and 1, highlighted above. The values are as follows:

1	Channel 0 0 2 (Shear x2) (Twist x3) (Extends x4)
2	0.000000 0.000000
3	0.000000 0.000000 0.000000
4	0.000000 0.000000 1.000000 1.000000
5	6

### Line 1

- Channel 0 is the main view channel. The first number indicates it is channel No. 0. The channel numbers must start from 0 and grow incrementally.
- The second number indicates this channel is rendered on visual server No. 0, which is always the case for the MiniSim.
- The third number indicates the coordinates of the channel span with regard to display screen No. 0. **This is usually the Windows display number minus 1.** This must be the same for all the subsequent Channels. See the *alternative method* to determine this number.
- The rest of the line is ignored, and is a key as to the settings in the following lines

### Line 2:

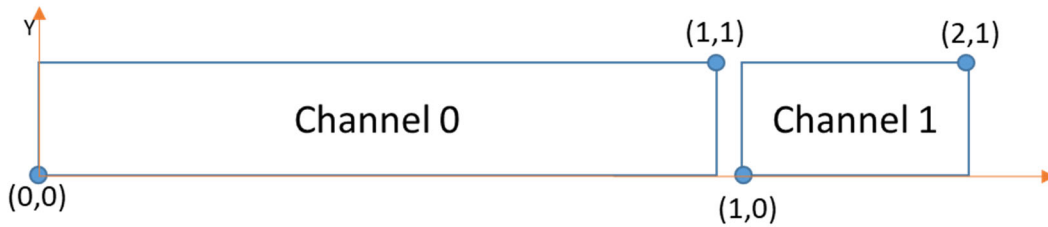
The second line in the block are x shear and y shear values for the channel, which will be applied to all viewports in the channel. The coordinate system in which the shears and twists are defined is the OpenGL coordinate system, with +x axis pointing toward the right, +y axis pointing up, and +z axis pointing out of the display screen and toward the viewer. A positive x shear moves the intersection point of the forward direction of the viewing camera and the view clipping planes to the right by shifting the view frustum to the left, thus rendering the scene to the left of the regular view, which has the appearance that the whole scene shifts to the right. A positive y shear moves the intersection point of the forward direction of the viewing camera and the view clipping planes toward the top by shifting the view frustum down, thus rendering the scene to the bottom of the regular view, which has the appearance that the whole scene shifts up. For example, a visible impact of a negative y shear is that the horizon is shifted downwards. To shift the horizon from the center of the view to the bottom edge, the y shear value needs to be set to -0.5. Similarly, to shift the rendering content by half the width of the view to the left, the x shear value needs to be set to -0.5.

### Line 3:

The third line are x twist, y twist, and z twist values for the channel, which will be applied to all viewports in the channel. A positive x twist rotates the viewing camera counterclockwise around the x axis, which raises the camera, thus showing more sky. A positive y twist rotates the viewing camera counterclockwise around the y axis, which turns the camera to the left, thus showing more content on the left side. A positive z twist rotates the viewing camera counterclockwise around the z axis, which raises the left side of the horizon. Twists are useful if the display screen is twisted.

### Line 4:

The fourth line contains the xmin, ymin, xmax and ymax coordinates of the channel span with regard to the displays as shown in the Windows screen resolution dialog. In the above case they are with regard to the display screen No. 0, or the left most display shown in the Windows dialog, and the 0, 0, 1, 1 range indicates the channel covers the full display:



**Line 5:**

The fifth line indicates how many viewports are contained in the channel. A viewport is a rectangular area that covers a portion of the rendering window that represents the channel. The viewport count can optionally be followed by a string, which is used as the name of the channel.

In this case there are 6 viewports in Channel 0 (Left View, Center View, Right View, Left Mirror, Center Mirror, and Right Mirror), and no viewports in Channel 1, which is the instrument panel.

**Viewport Blocks**

The viewport blocks are highlighted above. The viewports match the rendered images with the physical layout of the displays with respect to the driver’s eyepoint. Changing the display size and location will necessitate editing the viewports in the CCF file.

This example has 6 viewports in Channel 0, and no viewports in Channel 1. The first three correspond to the Left, Center, and Right viewports used to render the scene on each of the front three displays.

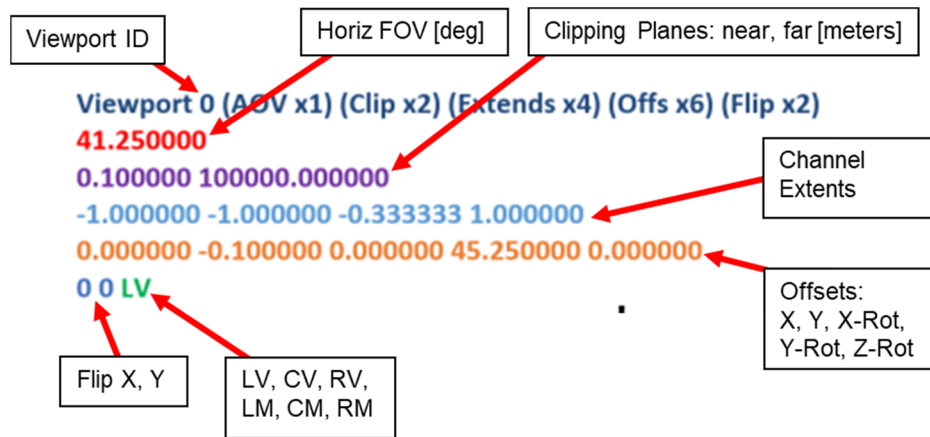
The other three are for the Left, Center, and Right mirrors. Only the **center mirror** is user-adjustable. If the extents of this viewport are changed, then the mirror will move and the ‘housing’ around the mirror viewport will re-scale along with the viewport. This is handy when you want to move the center mirror to the center display or reposition it, as on a single display system.

Taking the first viewport as an example:

1	Viewport 0 (AOV x1) (Clip x2) (Extends x4) (Offs x6) (Flip x2)
2	46.60
3	0.300000 35000.000000
4	-1.000000 -1.000000 -0.333333 1.000000
5	0.000000 -1.10000 0.000000 47.870000 0.000000
6	0 0 LV

**Line 1:**

The first line indicates it is viewport No. 0 for the channel. Just like channel numbers, viewport numbers for each channel must start with 0 and grow incrementally, and the total number of viewports must match the number in the fifth line of the channel description block. The rest of the line is ignored and contains a key as to the meaning of the other lines.

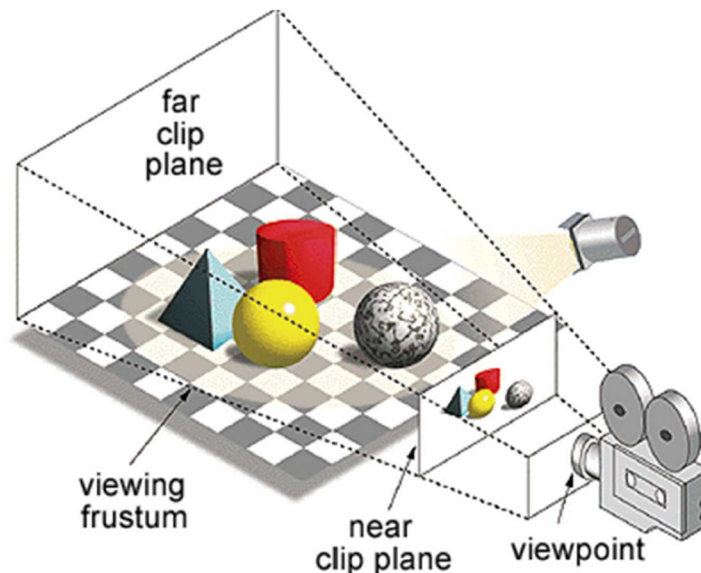


**Line 2:**

The second line specifies the horizontal field of view of the viewport in degrees. This is usually calculated by creating a scale drawing of the monitor locations with respect to the driver's eyepoint. Optionally, a second number can be used to specify the vertical field of view. If it is absent, the vertical field of view will be calculated automatically based on the aspect ratio of the viewport defined in the go.cmd (in this case, its **wide**).

**Line 3:**

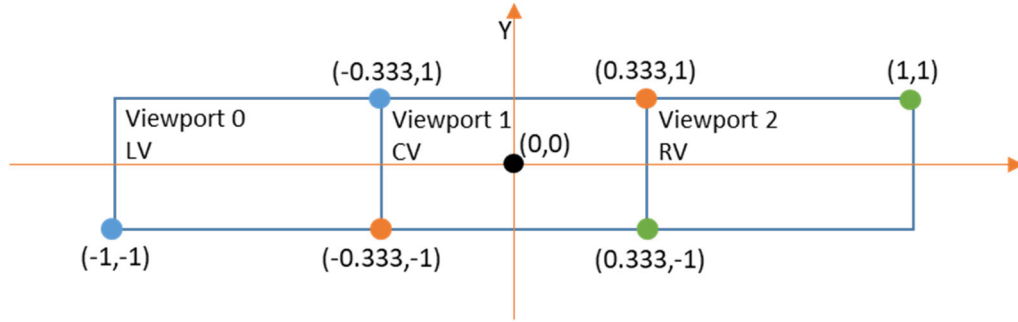
The third line specifies the near and far clipping planes of the view frustum for the viewport in **meters**. Even though the MiniSim scenarios use feet as the unit, it is converted to meters for rendering. Only 3D objects between the clipping planes is rendered.



**Line 4:**

The fourth line in the viewport description block contains the Xmin, Ymin, Xmax and Ymax coordinates of the viewport span within the channel. Please note that they have a range of -1 to

1, instead of 0 to 1 like the channel span. (Black dot in next figure is the center for the channel, e.g. could extend the CM by changing lower and upper coordinates)



**Line 5:**

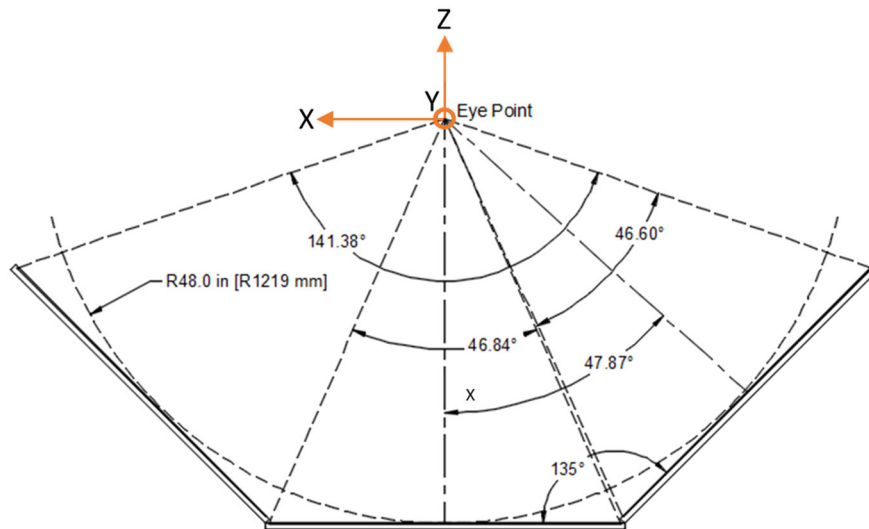
The fifth line in the viewport description block are x shear, y shear, x twist, y twist and z twist values for the viewport. The viewport twist and shear values are added to the channel twist and shear values to determine the final twist and shear values for the viewport.

**Line 6:**

The sixth and last line indicates whether the rendering image should be flipped along the x axis and the y axis. For viewports representing mirror views, the images need to be flipped along the x axis. Following the x and y flip values is a string that is used for the viewport's name. The name of a viewport is very important in determining the functionality of the viewport. Only LV, CV, RV, LM, CM, RM are used.

The viewports for the LV, CV, RV for 48" diagonal monitors are shown below, as viewed from above. The driver's eyes are 48" from the screen, and the angle between displays is 135 deg. No twist is required as all the displays are tangent to the 48" radius. The data for this setup are as follows:

Name	Code	Viewport	Horiz FOV [deg]	Y-rotation [deg]
Left View	LV	0	46.6	47.87
Center View	CV	1	46.84	0
Right View	RV	2	46.6	-47.87

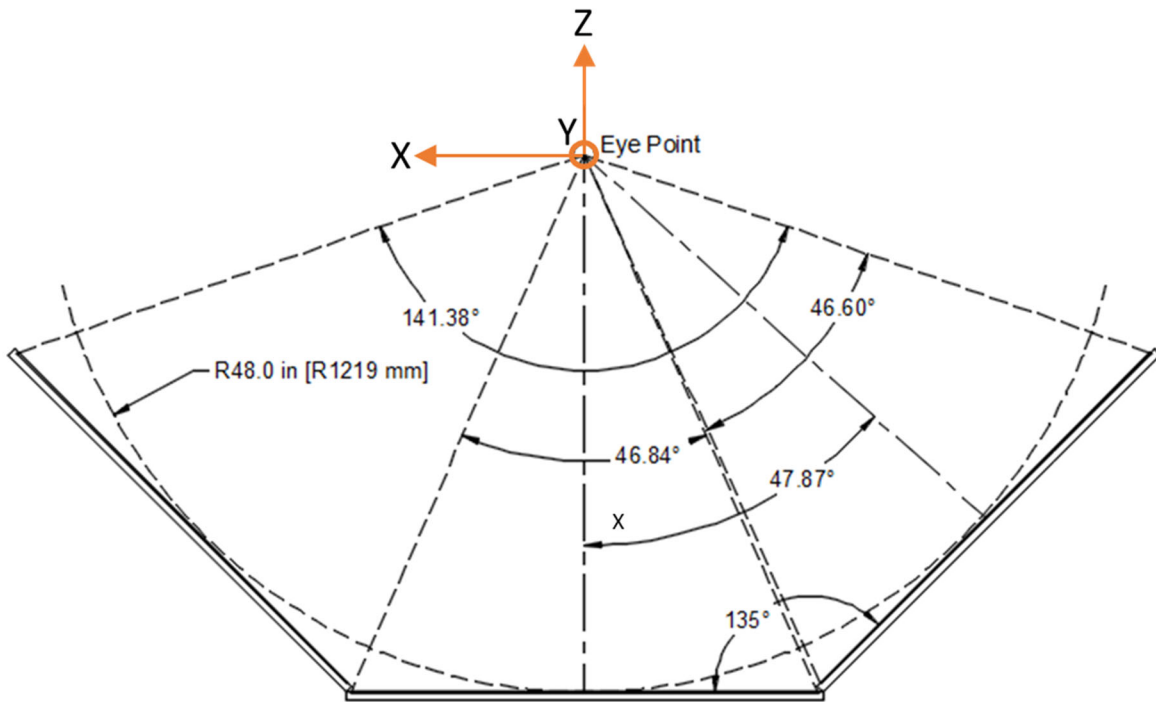


Display Geometry (plan view)  
 Quarter-Cab/Simplified-Cab MiniSim  
 48" LED LCD Displays, at 45.0 Degrees  
 ~140 Degree Horizontal FOV

## Viewport Settings

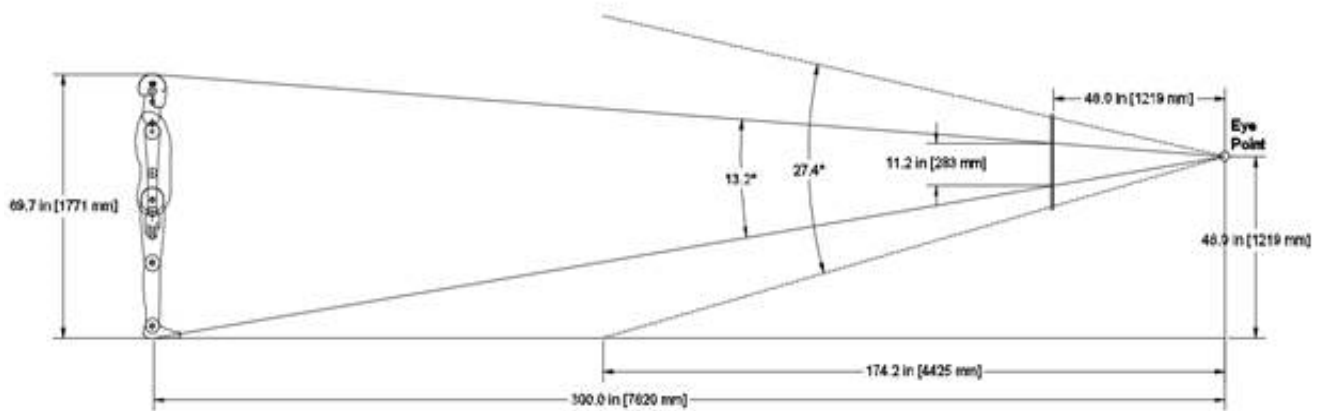
### Viewport Background:

In general, the goal is to represent the objects on the screen at 100% scaling with no distortion when possible. With larger screens (say 48" diagonal), this is relatively easy to do. The diagram below shows the viewport geometry for the 48" LCD screens located 48" from the eyepoint. The angle between screens here is 135 degrees, with a total horizontal field of view of about 141 degrees.



Display Geometry (plan view)  
 Quarter-Cab/Simplified-Cab MiniSim  
 48" LED LCD Displays, at 45.0 Degrees  
 ~140 Degree Horizontal FOV

When the above geometry is used to create viewports to render the scene and display it on the same 48" monitors, the result is a scene scaled at 100%. Also, the viewports are orthogonal to and centered on the eyepoint (in the horizontal plane). The results is that for a scene with a 69.7" pedestrian rendered 300' from the driver, the pedestrian on the screen should be 11.2" tall, below. The effect is like viewing the world through windows that are the exact size and location of the monitor screens.



The problem comes in when smaller monitors are used. If the viewport scaling is kept at 100% with small screens, the effect is like viewing the world through much smaller windows; the driver will see less of the

environment and may find it hard to see signs, signals, etc. Our compromise is to set-up 'desktop' systems with 24" displays with viewports consistent with 42" displays. We feel this works well for many uses and results in a system that most people find drivable and comfortable, but this may or may not be an issue for your specific application.

If you want us to work out the correct viewport settings for your system, I will need some data from you.

- What size monitors do you have now (or make and model)?
- What is the angle between the monitors? I measure with a tool like this:



- How far are they from the typical driver's eyepoint? I measure from the temple to the front screen with a tape measure.

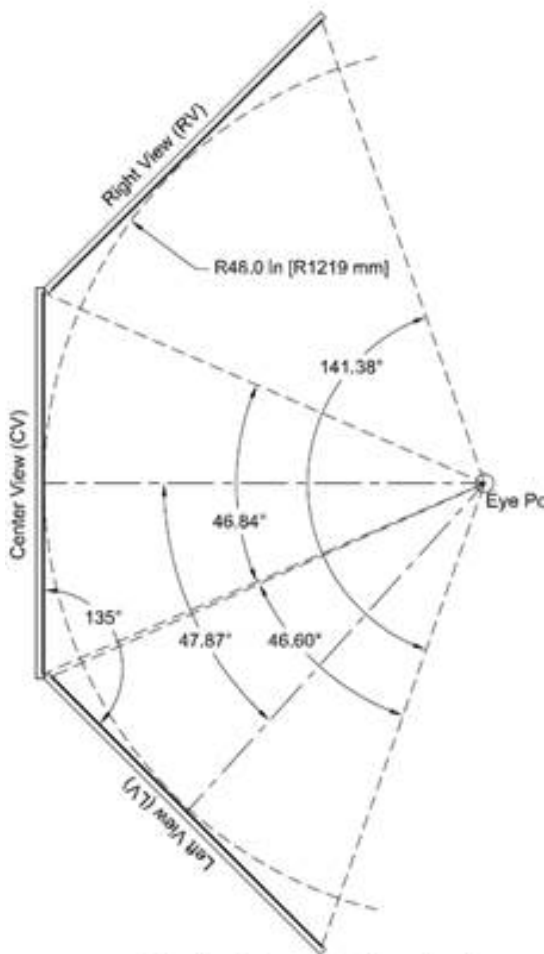
The viewport setting is a 'system' characteristic and as such it is specified in a configuration file, called the Channel Configuration File (.cf). Typically the same viewports are used for all drivers and scenarios once its configured.

If you get the viewport settings exactly right, then when seated in the simulator, the images on the screens will match up very well. Sitting in the simulator, objects and lane lines that cross between displays should line up very well. Small differences where the screens meet may actually be due to the physical alignment of the monitors – remember the viewports assume the monitors match up exactly. It can take some patient adjusting and fiddling with the mounts and leveling feet to get right, but it is worth it!



## miniSim Viewports for Three 48" Displays

As described previously, the typical miniSim displays look something like this, with the highlighted lines containing the viewport settings for the 48" miniSim display setup.



Display Geometry (top view)  
 Quarter-Cab/Simplified-Cab MiniSim  
 48" LED LCD Displays, at 45.0 Degrees  
 ~140 Degree Horizontal FOV

```

Channel 0 0 2 (Shear x2) (Twist x3) (Extends x4)
0.000000 0.000000
0.000000 0.000000 0.000000
0.000000 0.000000 1.000000 1.000000
6
Viewport 0 (AOV x1) (Clip x2) (Extends x4) (Offs x6) (Flip x2)
46.60
0.300000 35000.000000
-1.000000 -1.000000 -0.333333 1.000000
0.000000 -1.100000 0.000000 47.870000 0.000000
0 0 LV
Viewport 1 (AOV x1) (Clip x2) (Extends x4) (Offs x6) (Flip x2)
46.84
0.300000 35000.000000
-0.333333 -1.000000 0.333333 1.000000
0.000000 -1.100000 0.000000 0.000000 0.000000
0 0 CV
Viewport 2 (AOV x1) (Clip x2) (Extends x4) (Offs x6) (Flip x2)
46.60
0.300000 35000.000000
0.333333 -1.000000 1.000000 1.000000
0.000000 -1.100000 0.000000 -47.870000 0.000000
0 0 RV
Viewport 3 (AOV x1) (Clip x2) (Extends x4) (Offs x6) (Flip x2)
42.000000
0.700000 100000.000000
0.346667 0.555000 0.611667 0.865000
0.000000 0.000000 0.000000 0.000000 0.000000
1 0 CM
Viewport 4 (AOV x1) (Clip x2) (Extends x4) (Offs x6) (Flip x2)
27.000000
0.700000 100000.000000
-0.632000 -0.995000 -0.438000 -0.449000
0.000000 0.000000 0.000000 0.000000 0.000000
1 0 LM
Viewport 5 (AOV x1) (Clip x2) (Extends x4) (Offs x6) (Flip x2)
25.000000
1.100000 100000.000000
0.894333 -0.714000 0.984000 -0.346000
0.000000 0.000000 0.000000 0.000000 0.000000
1 0 RM
Channel 1 0 2 (Shear x2) (Twist x3) (Extends x4)
0.000000 0.000000
0.000000 0.000000 0.000000
1.000000 0.000000 2.000000 1.000000
0 IP
    
```

## Non-Standard Viewport Setup Example

This example is based on measurements for a miniSim install on a cab supplied by Sim Gear.



### Measurements

The first step was to measure the monitor dimensions:

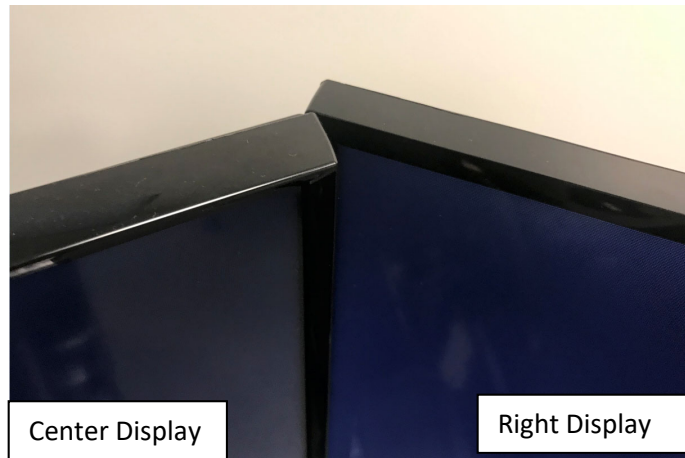
Overall: 37.75" Wide x 20.75" Tall x 0.75" Deep

Display Area: 34 15/16" Wide x 19 11/16 Tall

The angles between the monitors were measured with a goniometer, and were both exactly 125 degrees.

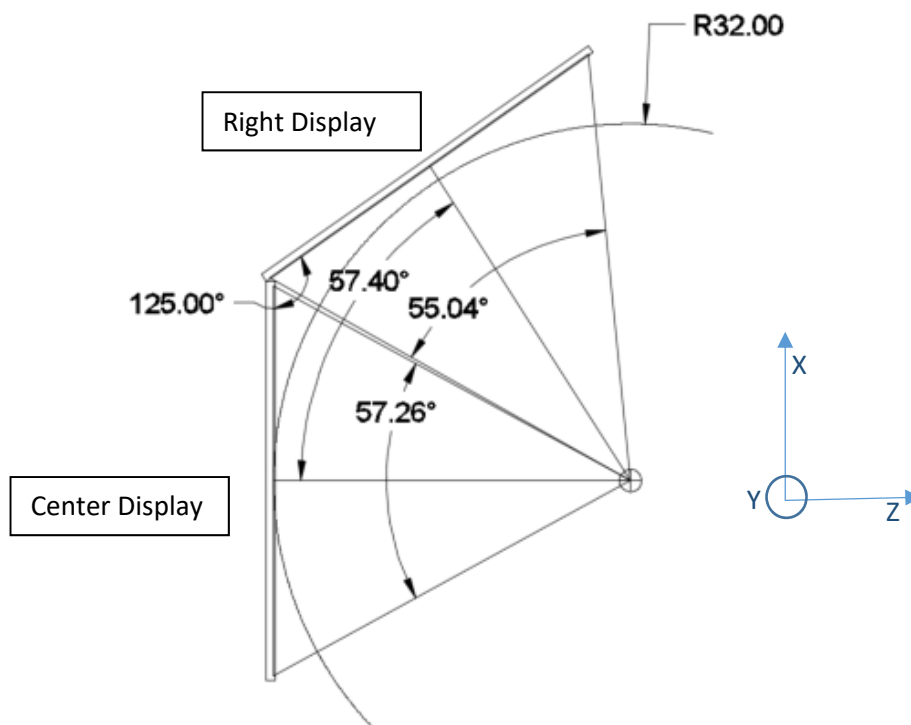
The distance between the front display and the driver's temple was 32"

The monitors were adjusted so that the front monitor overlapped the side monitors by the width of the bezel. The mounts were adjusted so the monitors met squarely, and overlap was consistent from top to bottom. There are only the front 3 channels and the operator display. There **is not** a discrete instrument panel display, instead, when the instruments are 'off', virtual instruments are rendered on the center screen.



### Calculation

These measurements were used to create a scale drawing in a CAD program:



The field of view for each monitor was calculated based on the monitor display width. The angle between the monitor centerlines was also calculated.

The go.cmd file for this system looks like this:

```

set MININADSCOMMMODE=PSEXEC
set DYNA=car
set NUMCHN=1
set INSTRUMENTS=off
set FORMAT=wide
set INSETVIEW=off
set SIDEMIRRORS=on
set MININADSDAQ=on

```

Based on these settings, the miniSim will read the following channel configuration file (.cfc) and use its contents for setting up the rendering channels and viewports. In this case, the miniSim will open this file:

`\bin.x64\mininads_1ch.wide.sidemirrors.noinst.cfc`

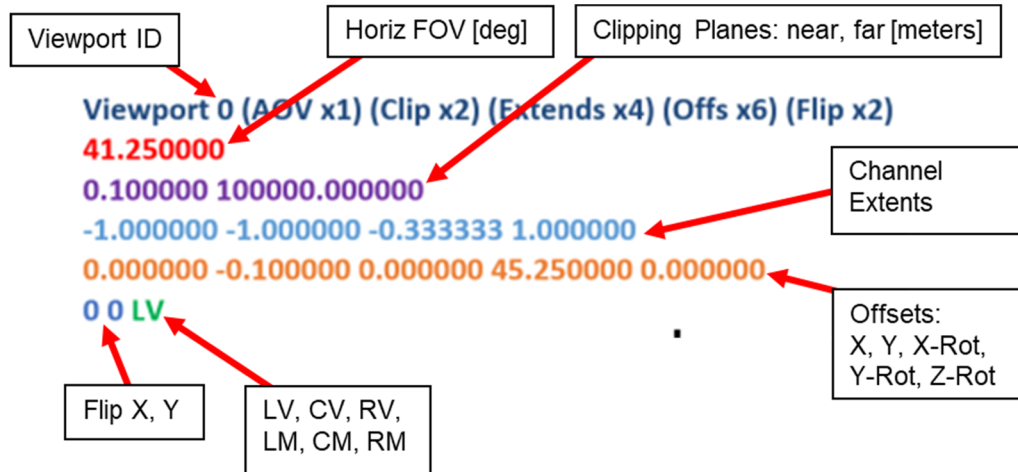
The contents of the .cfc file look like this. Note there is **NO** instrument panel display, so there is only a Channel 0, and no Channel 1. This is a **one channel** system. This .cfc has the viewports for the LV, CV, RV monitors, **highlighted** below.

```
Channel 0 0 2 (Shear x2) (Twist x3) (Extends x4)
0.000000 0.000000
0.000000 0.000000 0.000000
0.000000 0.000000 1.000000 1.000000
6
Viewport 0 (AOV x1) (Clip x2) (Extends x4) (Offs x6) (Flip x2)
55.04
0.100000 100000.000000
-1.000000 -1.000000 -0.333333 1.000000
0.000000 -1.100000 0.000000 57.400000 0.000000
0 0 LV
Viewport 1 (AOV x1) (Clip x2) (Extends x4) (Offs x6) (Flip x2)
57.26
0.100000 100000.000000
-0.333333 -1.000000 0.333333 1.000000
0.000000 -1.100000 0.000000 0.000000 0.000000
0 0 CV
Viewport 2 (AOV x1) (Clip x2) (Extends x4) (Offs x6) (Flip x2)
55.04
0.100000 100000.000000
0.333333 -1.000000 1.000000 1.000000
0.000000 -1.100000 0.000000 -57.400000 0.000000
0 0 RV
Viewport 3 (AOV x1) (Clip x2) (Extends x4) (Offs x6) (Flip x2)
42.000000
0.700000 100000.000000
0.346667 0.555000 0.611667 0.865000
0.000000 0.000000 0.000000 0.000000 0.000000
1 0 CM
Viewport 4 (AOV x1) (Clip x2) (Extends x4) (Offs x6) (Flip x2)
27.000000
0.700000 100000.000000
-0.632000 -0.995000 -0.438000 -0.449000
0.000000 0.000000 0.000000 0.000000 0.000000
1 0 LM
Viewport 5 (AOV x1) (Clip x2) (Extends x4) (Offs x6) (Flip x2)
25.000000
1.100000 100000.000000
0.894333 -0.714000 0.984000 -0.346000
0.000000 0.000000 0.000000 0.000000 0.000000
1 0 RM
```

The data from the example drawing above are as follows:

Name	Code	Viewport	Horiz FOV [deg]	Y-rotation [deg]
Left View	LV	0	55.04	57.4
Center View	CV	1	57.26	0
Right View	RV	2	55.04	-57.4

The values are entered into the viewport blocks as shown:



## Display Performance

Assuming you have all the images on the right screens, and the size looks good, next you need to check performance.

Frame Rate

Synch

Shear Lines

Jitter

# APPENDIX